

DIGITAL ETHERNET UNIBUS Network Adapter Technical Manual



DIGITAL ETHERNET UNIBUS Network Adapter Technical Manual

PRELIMINARY

Prepared by Educational Services
of
Digital Equipment Corporation

Preliminary Edition, December 1982

© Digital Equipment Corporation 1982
All Rights Reserved

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Printed in U.S.A.

This document was set on DIGITAL's DECset Integrated Publishing System.

The following are trademarks of Digital Equipment Corporation:

digital™

DEC
DECmate
DECset
DECsystem-10
DECSYSTEM-20
DECUS

DECwriter
DIBOL
MASSBUS
PDP
P/OS
Professional
Rainbow
RSTS

RSX
TS05
TSV05
UNIBUS
VAX
VMS
VT
Work Processor

CONTENTS

Page

CHAPTER 1 INTRODUCTION

1.1	SCOPE.....	1-1
1.2	ETHERNET OVERVIEW.....	1-1
1.3	DEUNA GENERAL DESCRIPTION.....	1-4
1.4	DEUNA SYSTEM OPERATION.....	1-6
1.4.1	ETHERNET Physical Channel Functions.....	1-6
1.4.2	ETHERNET Data Link Functions.....	1-9
1.4.3	Data Encapsulation.....	1-9
1.4.4	Data Decapsulation.....	1-9
1.4.5	Link Management.....	1-11
1.4.6	Diagnostics and Maintenance.....	1-12
1.5	DEUNA SPECIFICATIONS.....	1-12
1.6	RELATED DOCUMENTS.....	1-14

CHAPTER 2 PORT MODULE FUNCTIONAL DESCRIPTION

2.1	OVERVIEW.....	2-1
2.2	UNIBUS INTERFACE.....	2-3
2.2.1	DMA Control.....	2-3
2.2.1.1	RX DMA.....	2-4
2.2.1.2	Tll UNIBUS DMA.....	2-10
2.2.2	Port Control and Status Registers.....	2-13
2.2.2.1	Port Control and Status Register 0.....	2-13
2.2.2.2	Port Control and Status Register 1.....	2-18
2.2.2.3	Port Control and Status Register 2.....	2-20
2.2.2.4	Port Control and Status Register 3.....	2-21
2.3	MICROPROCESSOR SECTION.....	2-22
2.3.1	Microprocessor.....	2-22
2.3.2	Internal Registers.....	2-25
2.3.3	Default Station Address.....	2-26
2.3.4	Physical Address Registers.....	2-26
2.3.5	Port Switchpack Register.....	2-28
2.3.6	Timer.....	2-29
2.3.7	Internal Buses.....	2-29
2.4	LINK MEMORY CONTROL.....	2-29
2.4.1	Link Memory Arbitration.....	2-30
2.4.2	Link Transmit Address Counter.....	2-30
2.4.3	Link Receive Address Counter.....	2-33
2.4.4	Tll Addressing of Link Buffer Memory.....	2-35
2.4.5	Port-to-Link Interface.....	2-35
2.4.5.1	Link Memory Bus.....	2-35
2.4.5.2	Link Memory Address Control Signals.....	2-36
2.4.5.3	Command Register Control.....	2-36
2.4.5.4	Link Discrete Status.....	2-37
2.4.5.5	Clock and Reset.....	2-37

CONTENTS (Cont)

Page

CHAPTER 3 LINK MODULE FUNCTIONAL DESCRIPTION

3.1	INTRODUCTION.....	3-1
3.2	LINK MEMORY BUS.....	3-1
3.3	LINK REGISTERS.....	3-5
3.3.1	Command Register.....	3-5
3.3.2	Link Mode Register.....	3-7
3.3.3	Station Address RAM.....	3-10
3.4	PHYSICAL CHANNEL INTERFACE.....	3-12
3.4.1	Tranceiver Signals.....	3-12
3.4.2	Receiver.....	3-12
3.4.2.1	Receiver Squelch and Currier Sense.....	3-12
3.4.2.2	Manchester Decoder.....	3-13
3.4.2.3	Clock Shaper.....	3-13
3.4.2.4	Collision Squelch.....	3-13
3.4.3	Transmitter.....	3-13
3.4.3.1	Manchester Encoder.....	3-13
3.4.3.2	Transmit Enable Sync.....	3-14
3.5	TRANSMIT SECTION.....	3-14
3.5.1	Data Section.....	3-14
3.5.2	TX Data Latch.....	3-16
3.5.3	TX Message Byte Counter.....	3-16
3.5.4	TX Frame and Byte Sync.....	3-16
3.5.5	TX Shift MUX.....	3-16
3.5.6	TX Shifter.....	3-17
3.5.7	TX Output MUX.....	3-17
3.5.8	TX Status Information.....	3-17
3.5.9	Transmit State Machine.....	3-20
3.6	RETRY LOGIC.....	3-21
3.6.1	Collision Jam.....	3-21
3.6.2	Slot Time Counter.....	3-21
3.6.3	10 MHz Oscillator.....	3-21
3.6.4	Random Number Generator.....	3-21
3.6.5	Random Interval Mask/Latch.....	3-21
3.6.6	Interval Counter.....	3-22
3.6.7	Retry Counter.....	3-22
3.6.8	Retry State Machine.....	3-22
3.6.9	Time Domain Reflectometry.....	3-23
3.7	RECEIVE SECTION.....	3-23
3.7.1	Data Section.....	3-23
3.7.2	Receive MUX.....	3-25
3.7.3	Receive Shifter.....	3-25
3.7.4	RX Data Latch.....	3-26
3.7.5	RX Frame and Byte Sync.....	3-26
3.7.6	RX Byte Counter.....	3-26
3.7.7	Receive State Machine.....	3-26
3.7.8	Interpacket Delay.....	3-27
3.8	STATION ADDRESS DECODE.....	3-27
3.8.1	Physical/Logical Address Detection.....	3-28

CONTENTS (Cont)

	Page
3.8.2 Promiscuous Mode.....	3-28
3.8.3 Enable All Multicast.....	3-28
3.9 CRC LOGIC.....	3-28
3.10 TX/RX STATUS.....	3-29
3.11 LINK MEMORY.....	3-29
3.12 LINK MEMORY BUS CONTROLLER.....	3-30

CHAPTER 4 MICROCODE

4.1 OVERVIEW.....	4-1
4.2 STRUCTURE.....	4-1
4.3 SUPERVISOR.....	4-1
4.3.1 Initialization.....	4-1
4.3.2 Scheduling.....	4-2
4.3.3 Datagram Receive Process.....	4-3
4.3.4 Command Execution Process.....	4-7
4.3.4.1 Port Commands.....	4-7
4.3.4.2 Ancilliary Commands.....	4-9
4.3.5 Timer Process.....	4-10
4.3.6 Loop and Maintenance Process.....	4-10
4.3.7 Transmit Datagram Process.....	4-14
4.3.8 Null Process.....	4-15

FIGURES

Figure No.	Title	Page
1-1	Typical Large-Scale ETHERNET Configuration.....	1-3
1-2	DEUNA to ETHERNET Connection.....	1-5
1-3	PDP-11 Host System Block Diagram.....	1-7
1-4	VAX-11 Host System Block Diagram.....	1-8
1-5	Format of an ETHERNET Data Packet.....	1-10
2-1	PORT Module Functional Block Diagram.....	2-2
2-2	DMCSR Bit Format.....	2-5
2-3	DMAT Bit Format.....	2-7
2-4	DMAF Bit Format and Descriptions.....	2-8
2-5	DMWC Register Bit Format and Bit Descriptions.....	2-9
2-6	Microprocessor DMA Address Register Bit Format and Description.....	2-11
2-7	Data Register MDMDR0.....	2-12
2-8	Microprocessor DMA Data Register MDMDR1.....	2-12
2-9	PCSR0 Format.....	2-13
2-10	PCSR1 Format.....	2-18
2-11	PCSR2 Format.....	2-20
2-12	PCSR3 Format.....	2-21

FIGURES (Cont)

Figure No.	Title	Page
2-13	Tll Address Space.....	2-24
2-14	Physical Address Register Bit Configuration.....	2-27
2-15	Port Switchpack Register Bit Configuration.....	2-28
2-16	LTAC Configuration.....	2-31
2-17	LTAC Bit Configuration.....	2-32
2-18	Receive Address Counter Configuration.....	2-33
2-19	LRBAF, LCBAF Bit Configuration.....	2-34
3-1	Link Module Functional Block Diagram.....	3-2
3-2	Format of Link Command Register.....	3-6
3-3	Link Mode Register Format.....	3-8
3-4	Station Address RAM Format.....	3-11
3-5	Transmit Buffer Format Before Transmission.....	3-15
3-6	Transmit Buffer Format.....	3-18
3-7	Receive Buffer Format.....	3-24
4-1	Receive Flow Diagram.....	4-5
4-2	Receive DMA Done Flow Diagram.....	4-6
4-3	Port Command Processes.....	4-8
4-4	Loop Process Flow Diagram.....	4-11
4-5	Station ID Flow Diagram.....	4-13
4-6	Transmit Flow Diagram.....	4-16
4-7	Transmit Done Flow Diagram.....	4-17

TABLES

Table No.	Title	Page
1-1	DEUNA Specifications.....	1-13
1-2	Related Hardware and Software Documents.....	1-14
2-1	DMCSR Bit Descriptions.....	2-6
2-2	PCSR0 Bit Descriptions.....	2-14
2-3	PCSR1 Bit Descriptions.....	2-19
2-4	PCSR2 Bit Descriptions.....	2-21
2-5	PCSR3 Bit Descriptions.....	2-22
2-6	Tll Interrupts.....	2-23
2-7	Internal Register Address Assignments.....	2-25
2-8	Port Switchpack Register Bit Descriptions.....	2-29
3-1	Memory Bus Signals.....	3-3
3-2	Discrete Control Bus Signals.....	3-3
3-3	Discrete Status Bus Signals.....	3-4
3-4	Clock Signal.....	3-5
3-5	Link Command Register Bit Descriptions.....	3-7
3-6	Bit Descriptions for Link Mode Register.....	3-9
3-7	Station Address RAM Bit Descriptions.....	3-12
3-8	Transmit Buffer Bit Descriptions.....	3-16
3-9	Transmit Status Bit Descriptions.....	3-19
3-10	Receive Buffer Status and Data Bit Description.....	3-25
4-1	Priority of Processes.....	4-3

1.1 SCOPE

This chapter provides an introduction to the DIGITAL ETHERNET UNIBUS Network Adapter (DEUNA). A brief overview of the ETHERNET local area network is included, followed by a description of the DEUNA, its operation, and specifications. Additional documents related to this manual are listed for the reader who wishes more information about the ETHERNET, the DEUNA, or local area networks.

1.2 ETHERNET OVERVIEW

The ETHERNET is a local area network that provides a communications facility for high-speed data exchange among computers and other digital devices located within a moderately sized geographic area. It is intended primarily for use in such areas as office automation, distributed data processing, terminal access, and other situations requiring economical connection to a local communication medium carrying traffic at high-peak data rates.

The primary characteristics of ETHERNET include:

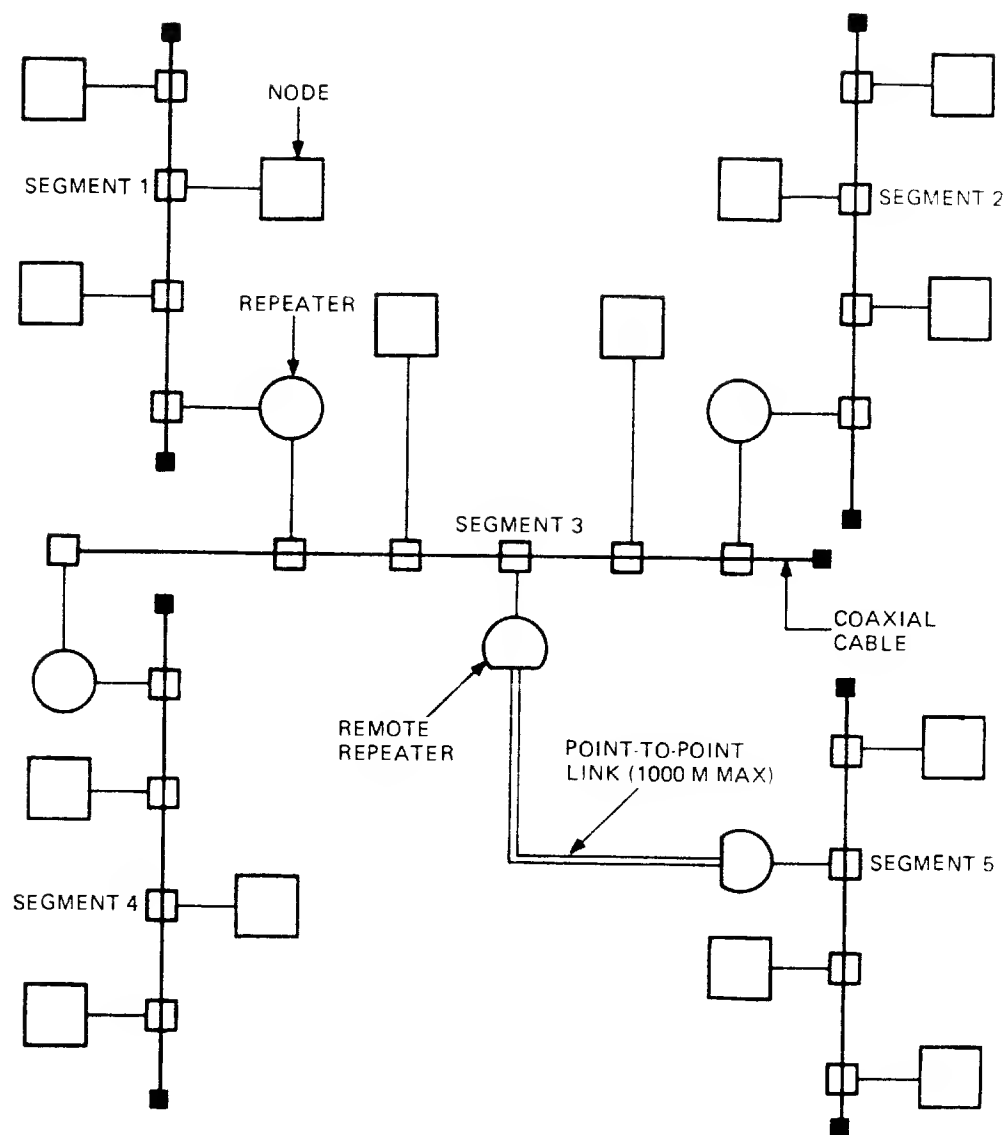
Topology	Branching bus.
Medium	Shielded coaxial cable, Manchester encoded digital base-band signaling.
Data Rate	10 million bits per second.
Maximum Separation of Nodes	2.8 kilometers (1.74 miles).
Maximum Number of Nodes	1,024
Network Control	Multiaccess -- fairly distributed to all nodes.
Access Control	Carrier Sense, Multiple Access with Collision Detect (CSMA/CD).
Allocation	Packet length from 64 to 1518 bytes (includes variable data field of from 46 to 1500 bytes).

The ETHERNET, like other local area networks, falls into a middle ground between long-distance, low-speed networks that carry data for hundreds or thousands of kilometers and specialized, very high-speed interconnections that are generally limited to tens of meters. Using a branching bus topology, ETHERNET provides a local area communications network allowing a 10M bits/s data rate over a coaxial cable at a distance of up to 2.8 km (1.74 mi).

A single ETHERNET can connect up to 1,024 nodes together for a local point-to-point/multipoint network. An example of a typical large-scale ETHERNET configuration is shown in Figure 1-1.

Rules for configuring ETHERNET are derived from certain limits that are imposed on the physical channel to ensure the optimal performance of the network. The maximum configuration for an ETHERNET is as follows:

- A segment of coaxial cable can be a maximum of 500 meters (1640.5 feet) in length. Each segment must be terminated at both ends in its characteristic impedance.
- Up to 100 nodes can be connected to any segment of the cable. Nodes on a cable segment must be spaced at least 2.5 meters (8.2 feet) apart.
- The maximum length of coaxial cable between any two nodes is 1,500 meters (4921.5 feet).
- The maximum length of the transceiver cable between a transceiver and a controller is 50 meters (164.05 feet).
- A maximum of 1,000 meters (3281 feet) of point-to-point link is allowed for extending the network.
- Repeaters can be used to continue signals from one cable segment of the ETHERNET to another. A maximum of two repeaters can be placed in the path between any two nodes.



TK-9817

Figure 1-1 Typical Large-Scale ETHERNET Configuration

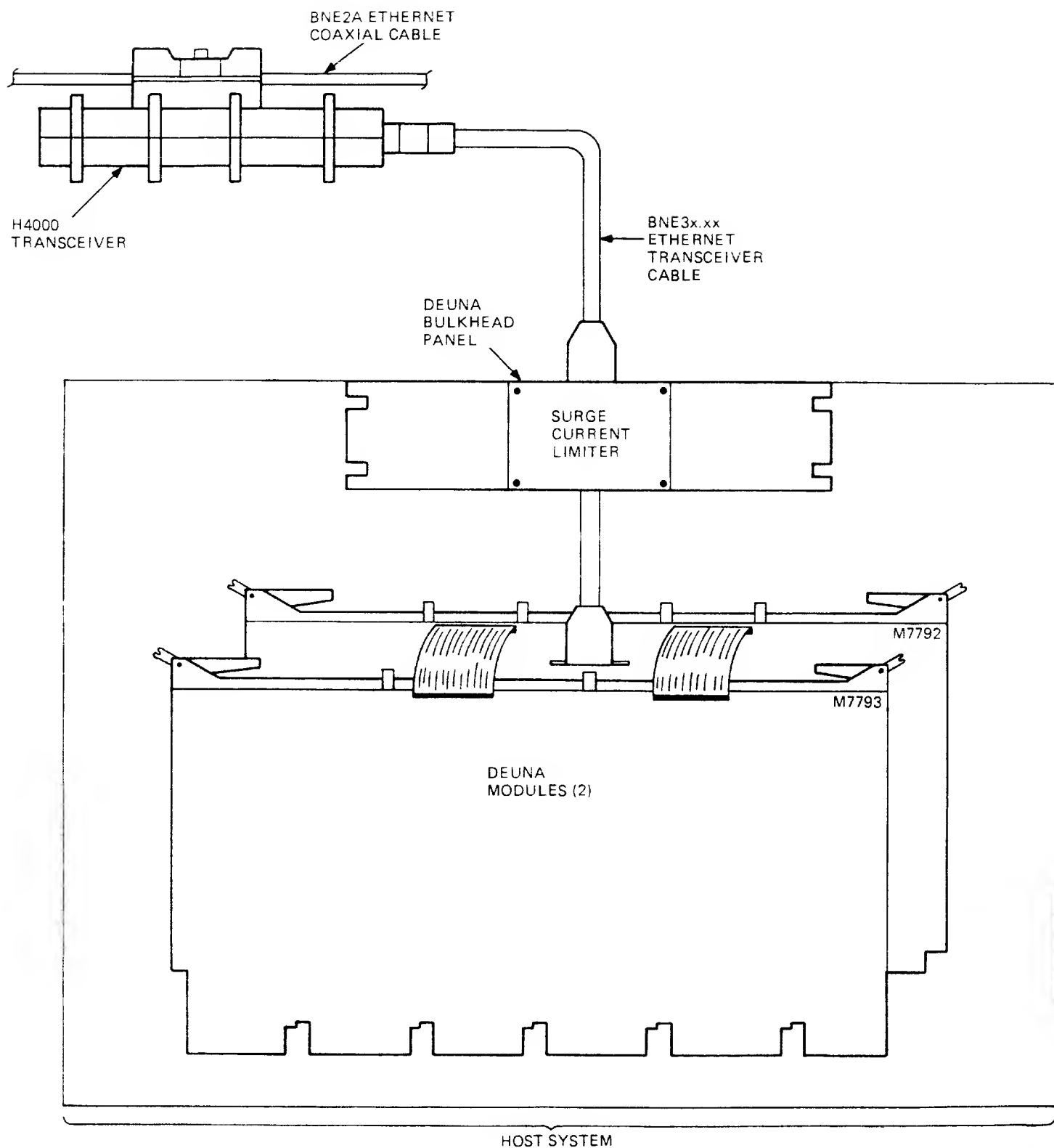
1.3 DEUNA GENERAL DESCRIPTION

The DEUNA is a data communications controller used to interface VAX-11 and PDP-11 family computers to the ETHERNET local area network. It complies with the ETHERNET specification and allows communication with up to 1024 addressable devices using the ETHERNET shielded coaxial cable.

Features of the DEUNA include:

- 10M bits/s transmission and reception,
- Transmit and receive data link management,
- Data encapsulation and decapsulation,
- Data encoding and decoding,
- Down-line loading and remote load detect capabilities,
- Internal ROM based microdiagnostics to facilitate diagnosis and maintenance of both the DEUNA and the DIGITAL H4000 transceiver,
- Collision detection and automatic retransmission,
- 32-bit Cyclic Redundancy Check (CRC) error detection, and
- 32 KB (16 KW) buffer for continuous datagram reception, transmission, and maintenance requirements.

The DEUNA is comprised of two hex-height modules, a bulkhead interconnect panel, and associated cables. It physically and electrically connects to the ETHERNET cable via the DIGITAL H4000 transceiver and the appropriate transceiver cable as shown in Figure 1-2.



TK-9818

Figure 1-2 DEUNA to ETHERNET Connection

1.4 DEUNA SYSTEM OPERATION

The DEUNA controller performs both the ETHERNET data link layer functions and a portion of the physical channel functions. It also provides the following network maintainability features.

- Loopback of data from other stations.
- Individual system identification.
- Loading and remote booting of UNIBUS systems from other stations on the network.

The DEUNA is a microprocessor based device that, when connected to the DIGITAL H4000 ETHERNET transceiver, provides all the logic necessary to connect VAX-11 and UNIBUS PDP-11 family minicomputers to an ETHERNET local area network (Figures 1-3 and 1-4). The controller microcode implements data encapsulation and decapsulation, data link management, and all channel access functions to ensure maximum throughput with minimum host processor intervention.

1.4.1 ETHERNET Physical Channel Functions

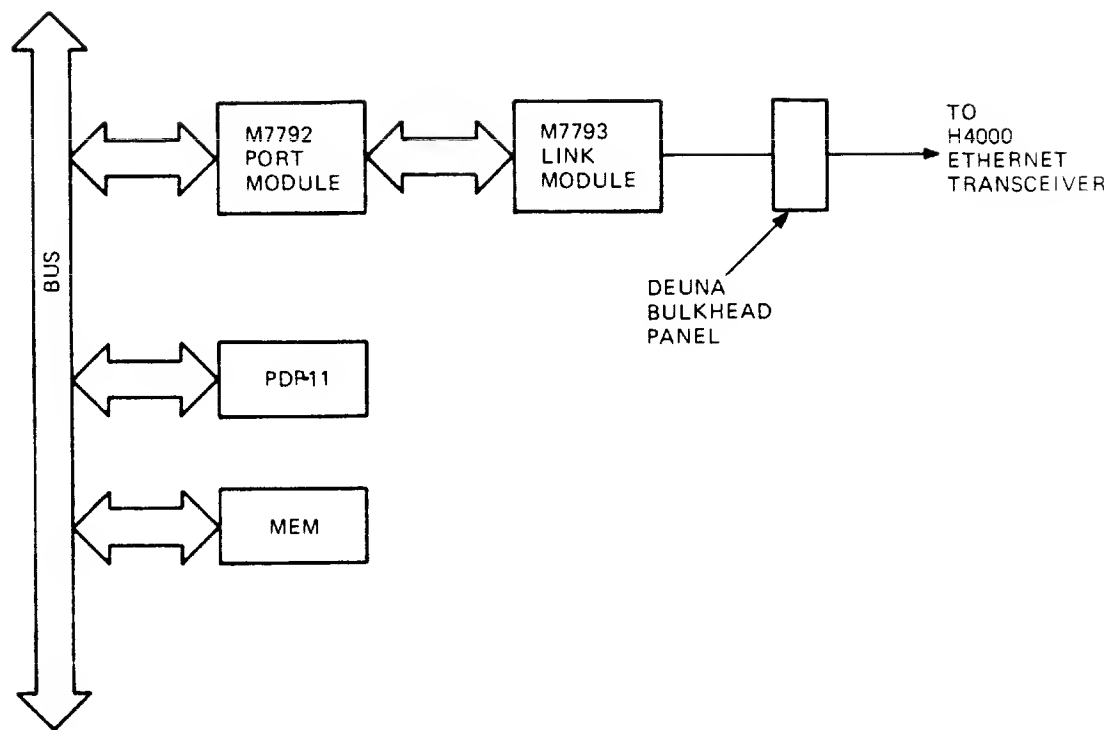
The DEUNA provides the following specific ETHERNET physical channel functions necessary to interface to the DIGITAL H4000 ETHERNET transceiver:

During Transmission

- Generates the 64-bit preamble for synchronization.
- Generates the Manchester encoding of data.
- Provides parallel-to-serial conversion of the frame.
- Ensures proper channel access by monitoring and sensing the carrier from any stations' transmission.
- Monitors the self-test collision detect signal from the DIGITAL H4000 transceiver.

During Reception

- Senses carrier from any stations' transmission.
- Provides serial-to-parallel conversion of the frame.
- Performs Manchester decoding of the incoming bit streams.
- Buffers received frames.
- Synchronizes to the preamble and removes it prior to processing.



TK-9816

Figure 1-3 PDP-11 Host System Block Diagram

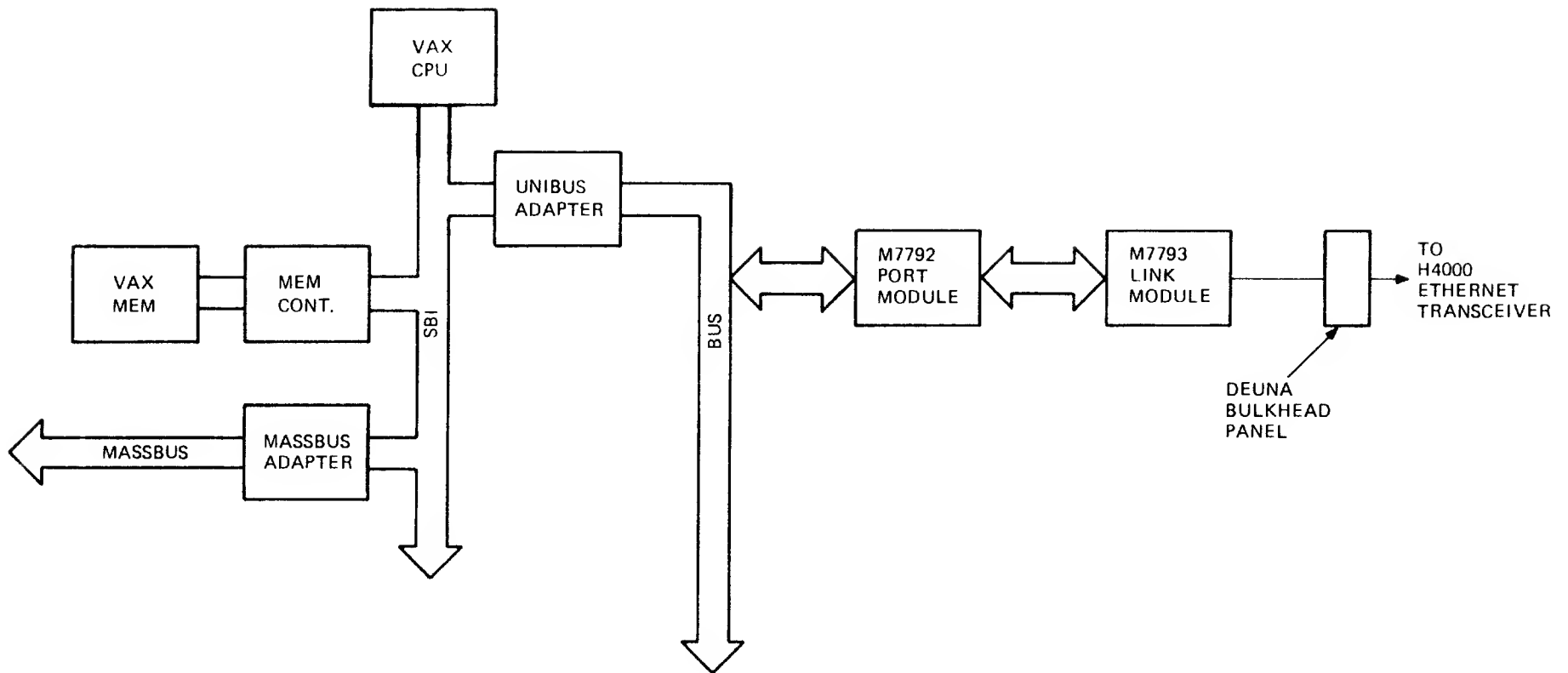


Figure 1-4 VAX-11 Host System Block Diagram

1.4.2 ETHERNET Data Link Functions

The DEUNA provides the following specific ETHERNET data link layer functions.

- Calculates the 32-bit CRC value and places it in the frame sequence field upon transmission
- Attempts automatic retransmission upon collision detection
- Checks incoming frames for proper CRC value
- Performs all address filtration

1.4.3 Data Encapsulation

The ETHERNET frame format for the transmission of data packets is shown in Figure 1-5. Each frame begins with a 64-bit preamble, that is used for synchronization by the receiving station, and ends with a 32-bit frame check sequence. Frames are separated by a specified minimum spacing period of 9.6 microseconds.

The destination address field contains the address(es) of the station(s) where the packet is sent. The address may represent: the physical or logical address of a particular station or group of stations; a multicast, or group address, associated with a set of stations; and a broadcast address for broadcast to all stations on the network.

The source address field specifies the physical address of the transmitting station. Each DEUNA has a unique 48-bit address value determined during manufacture. This value is called the default physical address. The system software can override this value and insert a more appropriate logical address into the source address field upon transmission.

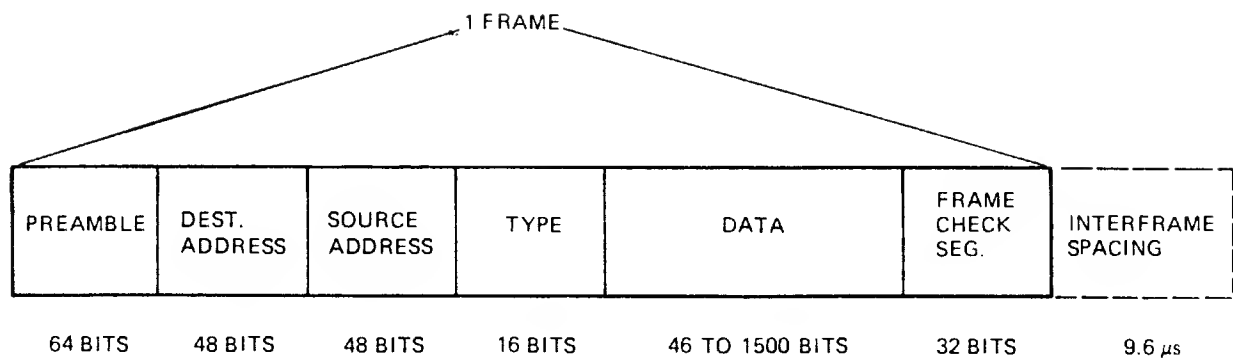
The type field is specified for use by high-level network protocols and it indicates how the content of the data field is to be interpreted. The type field indicates the higher level architecture that can further decapsulate the data.

The data field may have between 46 and 1500 bytes of data. The DEUNA can be initialized to automatically insert null characters if the amount of data is less than the minimum 46 byte data size.

The frame check sequence contains a 32-bit Cyclic Redundancy Check (CRC) value that is determined and inserted by the DEUNA during transmission.

1.4.4 Data Decapsulation

The DEUNA continuously monitors the signals transmitted by the DIGITAL H4000 transceiver. After sensing a carrier, the preamble sequence of the received frame is used by the controller for synchronization. It then processes the destination address field through a hardware comparator to determine whether or not the incoming frame is intended for its station. The DEUNA accepts only



TK-9814

Figure 1-5 Format of an ETHERNET Data Packet

frames that have a destination address that matches one of the following types of address.

1. The physical address of the station
2. The broadcast address for all stations
3. One of the 10 multicast group addresses that the user may assign to the DEUNA
4. Any multicast address
5. All addresses, when desired

The DEUNA performs a hardware comparison of the 48-bit destination address to determine if there is a match with the station's physical address or with one of the ten user designated logical multicast addresses. If necessary, all multicast addresses may be passed to higher level software for decoding when more than ten multicast address groups are required by the user.

To assist in network management functions and to aid in fault diagnosis, the DEUNA can operate in a mode that effectively disregards the internal address filter logic. This allows all frames received from the network to be accepted. The DEUNA verifies the integrity to the received data by recalculating the 32-bit CRC value and comparing it with the CRC that is obtained from the received frame.

1.4.5 Link Management

The method utilized by the ETHERNET for channel access is called carrier sense, multiple access with collision detect (CSMA/CD). The DEUNA controls all of the link management functions necessary to successfully place or remove a frame of data on the ETHERNET network. These functions include:

Carrier Deference	The DEUNA monitors the physical channel for traffic and when the channel is busy, refers to the passing frame by delaying any transmission of its own.
Collision Detection	Collisions occur when two or more controllers attempt to transmit data simultaneously on the channel. The DEUNA monitors the collision sense signal generated by the DIGITAL H4000 transceiver. When a collision is detected, the DEUNA continues to transmit to ensure that all network stations detect the collision.

Collision Backoff Retransmission

When a controller has attempted transmission and encountered a collision on the channel, it attempts a retransmission a short time later. The schedule for retransmission is determined by a controlled randomization process. The DEUNA attempts to transmit a total of sixteen times and reports an error if it is not successful.

1.4.6 Diagnostics and Maintenance

The DEUNA utilizes both microdiagnostics and extensive system and network diagnostics to greatly minimize the time to isolate and diagnose a network communication fault. On-board self-test microdiagnostics automatically perform a test of the major DEUNA component logic both upon powerup and at the user's discretion. Light emitting diodes on the edge of the port module (M7792) provide an indication of a specific module problem.

The DEUNA does not allow itself to transmit significantly longer than the maximum ETHERNET frame transmit period. It contains an automatic control to prevent monopolizing the ETHERNET channel. The controller can differentiate between normal frame collisions on the physical channel and cable shorts or cable opens. A built-in Time Domain Reflectometry (TDR) circuit is utilized to determine the type of cable fault and its approximate location.

The controller continuously monitors the power applied to the DIGITAL H4000 transceiver to ensure compliance with the transceiver requirements. In addition, the H4000 provides a positive functional verification (heartbeat) after every attempted transmission which indicates its proper operation, including the collision sense circuitry.

Comprehensive system diagnostics provide loopback capability through the DEUNA, transceiver, or the ETHERNET network itself. The DEUNA allows remote stations to loopback through it once the DEUNA has successfully passed the the on-board self-test microdiagnostic. This provides both a local and remote station diagnostic capability. Network error conditions are detected and statistics tabulated for use by higher level network management applications.

1.5 DEUNA SPECIFICATIONS

The DEUNA specifications are outlined in Table 1-1.

Table 1-1 DEUNA Specifications

Specification	Description
Performance	
Operating Mode	Half-duplex
Data Format	ETHERNET specification
Data Rate	10M bits/s
Network Specifications	1024 stations maximum
UNIBUS Conductor Loading	
Module Pair	4 dc loads 2 ac loads
DC Power Requirements	
Port Module	+5 V, 7.0 A
Link Module	+5 V, 9.0 A -15 V, 2.0 A (for H4000 transceiver)
Physical Size	
Port and Link Modules	Height (hex): 21.4 cm (8.4 in) Length: 39.8 cm (15.7 in)
Cable Interface Panel	Height: 10.6 cm (4.0 in) Length: 10.6 cm (4.0 in)
Transceiver Cables available in 5 m (16.4 ft), 10 m (32.8 ft), or 20 m (65.6 ft) lengths.	
BNE3A-XX	Low loss PVC jacket/straight connector
BNE3B-XX	Low loss PVC jacket/right angle connector
BNE3C-XX	Low loss TEFLON* jacket/straight connector
BNE3D-XX	Low loss TEFLON* jacket/right angle connector

*TEFLON is a trademark of Dupont de Nemours & Co., Inc.

Table 1-1 DEUNA Specifications (Cont)

Specification	Description
Operating Environment	
Temperature	5°C to 50°C (41°F to 122°F)
Relative Humidity	10 to 90% (noncondensing)
Wet Bulb Temperature	32°C (90°F) maximum
Altitude	Sea level to 2.4 km (8,000 ft)
Shipping Environment	
Temperature	-40°C to 0°C (-40°F to 151°F)
Relative Humidity	0 to 90% (noncondensing)
Altitude	Sea level to 9 km (30,000 ft)

1.6 RELATED DOCUMENTS

Table 1-2 provides a list of documents related to this manual.

Table 1-2 Related Hardware and Software Documents

Title	Document Numbers
<u>DEUNA User's Guide</u>	EK-DEUNA-UG
<u>H4000 Technical Description</u>	(TBS)
<u>The ETHERNET, A Local Area Network, Data Link Layer and Physical Layer Specifications</u>	AA-K759A-TK
<u>ETHERNET Installation</u>	(TBS)
<u>Introduction to Local Area Networks</u>	EB-22714-18
<u>PDP-11 Bus Handbook</u>	EB-17525

DIGITAL personnel may order hardcopy documents from:

Digital Equipment Corporation
444 Whitney Street
Northboro, MA 01532

Attn: Publishing and Circulation Services (NR03/W3)
Order Processing Section

Customers may order hardcopy documents from:

Digital Equipment Corporation
Accessories and Supplies Group
Cotton Road
Nashua, New Hampshire 03060

For information call: 1-800-257-1710

Information concerning microfiche libraries may be obtained from:

Digital Equipment Corporation
Micropublishing Group (BUO/E46)
12 Crosby Drive
Bedford, MA 01730

CHAPTER 2
PORT MODULE
FUNCTIONAL DESCRIPTION

2.1 OVERVIEW

The port module (M7792) is the microprocessor controlled interface between the UNIBUS bus and the link module of the DEUNA. The logic on the port module is divided into three basic functional areas.

1. UNIBUS Interface -- This section contains the UNIBUS transceiver, port control and status registers (PCSRs), DMA control, UNIBUS interrupt control logic, and UNIBUS control.
2. Microprocessor Section -- This section is made up of the T11 microprocessor, 8K words of ROM for microprogram storage, 4K words of writable control store (WCS), internal register address decode, and timer.
3. Link Memory Control -- This section contains the link memory arbitration logic, control for the 16K words of link memory and the port-to-link interconnect.

The port module is a hex-height module that is installed in a small peripheral controller (SPC) slot of a UNIBUS backplane.

A functional block diagram of the port module is shown in Figure 2-1. The letters in the lower right corner of each block of the diagram indicate the page in the engineering drawings where the logic for that block is located.

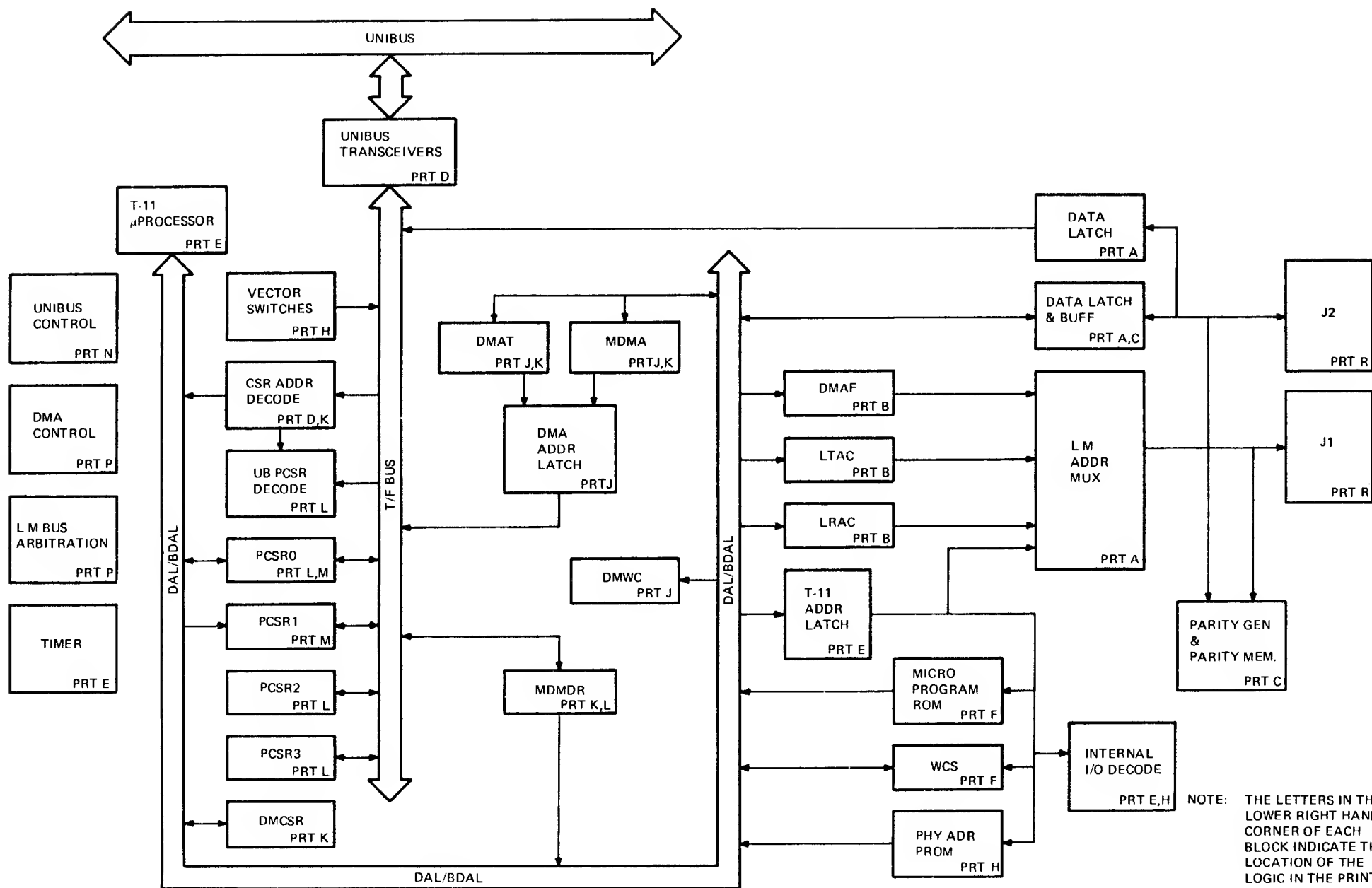


Figure 2-1 PORT Module Functional Block Diagram

2.2 UNIBUS INTERFACE

The UNIBUS interface logic on the port module is used to control the transfer of data between the host processor and the DEUNA. This logic generates the signals required of a bus master and bus slave on the UNIBUS. The DEUNA functions as bus master when data is to be transferred to or from the host processors memory via direct memory access (DMA). The DEUNA performs DMAs for the transfer of:

- Data received from the ETHERNET and
- Data to be transmitted on the ETHERNET.

The DEUNA functions as a bus slave when the host processor accesses the port control and status registers (PCSRs) for the transfer of control and status information.

NOTE

For a detailed explanation of UNIBUS architecture and protocol, refer to the PDP-11 Bus Handbook (EB-17525).

The port also controls the UNIBUS ACLO signal. It does this by setting a bit in the link mode register on the link module (see Section 3.3.2). This is used to get control of the host processor during a down-line load.

2.2.1 DMA Control

The DMA control logic is divided into two sections:

1. RX DMA -- Used when a message has been received from the ETHERNET and is ready to be transferred to the host processors memory.
2. T11 UNIBUS DMA -- Used when the T11 has to:
 - Read the ring structures in host memory,
 - Read data buffers in host memory for transmission on the ETHERNET, and
 - Write status information into the data buffers in host memory when the transmission is finished.

The control of each of these processes is implemented via programmed array logic (PAL) with the T11 UNIBUS DMA having a higher priority than the RX DMA process. This priority is established because the T11 UNIBUS DMA process transfers its data in smaller segments and therefore does not use the UNIBUS for long periods of time. This results in little effect on the RX DMA process and helps to maximize throughput.

A description of each of the PALs used in the DMA control is contained in the engineering drawings for the DEUNA.

2.2.1.1 RX DMA -- The DEUNA transfers received messages to host memory via the UNIBUS using DMAs. This is done asynchronous to the process or processes going on in the DEUNA. The port microprocessor (T11) starts the DMA transfers by loading a group of registers with the necessary address and word count information. Once this information is loaded, the T11 starts the DMA process by setting a bit in the DMA control and status register (DMCSR). This starts the DMA transfers under the control of the RX DMA PAL and the UNIBUS control logic.

The following registers, located on the port module, are used for the RX DMA process:

- DMCSR -- DMA control and status register,
- DMAT -- DMA-to-address register,
- DMAF -- DMA-from-address register, and
- DMWC -- DMA word count register.

The T11 controls the transfer of data from the buffers located on the link module to the host memory in the following manner:

1. The T11 loads the DMAT, DMAF, and the DMWC with the proper information.
2. The T11 sets the DMA GO bit in the DMCSR.
3. The DMA logic takes over and moves the data via NPRs to the host memory.
4. When all the data is transferred or when the RX DMA logic receives an error, it interrupts the T11.

The RX DMA logic only transfers words on the UNIBUS. The host software is responsible for throwing away the extra byte when transferring an odd byte buffer.

A description and layout of each of the registers used is given in the following sections.

2.2.1.1.1 DMA Control and Status Register (DMCSR) -- The DMCSR is used by the T11 to enable the DMA logic and to report DMA status to the T11. Figure 2-2 shows the DMCSR bit format and Table 2-1 gives a description of each of the bits.

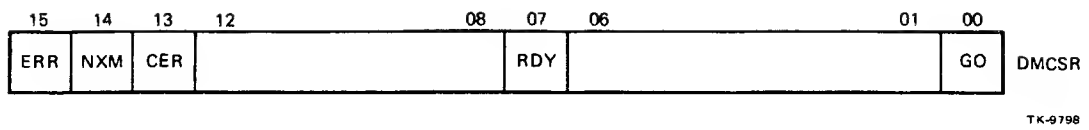


Figure 2-2 DMCSR Bit Format

Table 2-1 DMCSR Bit Descriptions

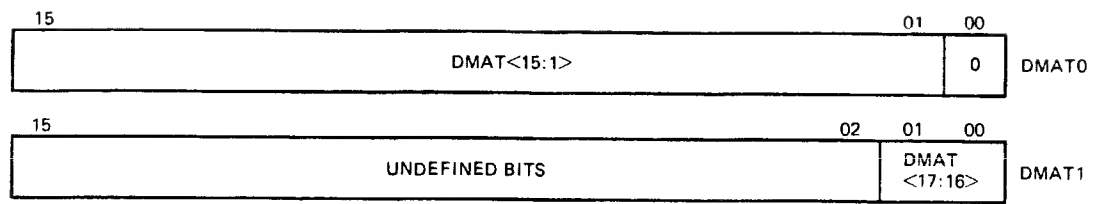
Bit	Field	Description
DMCSR<00>	GO	Go Bit -- This bit is set after the address and word count are loaded. On setting the DMA, the engine begins to arbitrate for the UNIBUS and starts data transfer to host memory.
DMCSR<07>	RDY	Ready Bit -- This bit creates an interrupt to the T11 to indicate that the word count has expired and the current DMA process is complete.
DMCSR<13>	CER	Collision Error -- When set indicates that the heartbeat from the H40000 or similar transceiver was not detected.
DMCSR<14>	NXM	Non-Existent Memory -- When set causes the DMA logic to interrupt the T11. Indicates a UNIBUS timeout to the address contained in the DMA-to-address register.
DMCSR<15>	ERR	DMA Logic Error -- Set when UPE or NXM are set.

2.2.1.1.2 DMA to Address Registers (DMAT0 and DMAT1) -- The DMAT registers are loaded by the T11 with the starting address of the receive buffer in host memory. DMAT0 contains the lower 16 bits of the address. DMAT1 contains the upper 2 bits of the address. These registers are a 17-bit counter that is incremented by two after each NPR cycle.

NOTE

Bit 0 of DMAT0 is always a 0. This is because the RX DMA logic only performs word transfers.

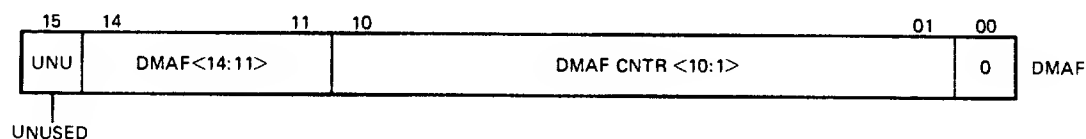
Figure 2-3 shows the format of each of the registers.



TK-9799

Figure 2-3 DMAT Bit Format

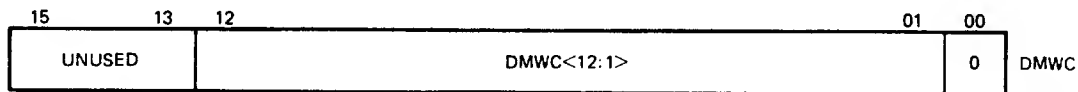
2.2.1.1.3 DMA-from-Address Register (DMAF) -- The DMAF register contains the receive buffer address in the link memory from which data is to be transferred. It is made up of a register for the upper four bits and a counter for the lower ten bits. The T11 loads the upper four bits from the link completed buffer address FIFO (refer to Section 2.4.3 for an explanation of the LCBAF). When the upper four bits are loaded, the lower ten bits are cleared. The counter is incremented by two after each NPR cycle. The address cannot overflow into the next buffer. Figure 2-4 shows the bit format of the DMAF register.



TK-9800

Figure 2-4 DMAF Bit Format and Descriptions

2.2.1.1.4 DMA Word Count Register (DMWC) -- The DMWC is loaded by the T11 with the number of words to be transferred to host memory by the DMA logic. The DMWC is implemented in a counter. After each NPR cycle it is decremented by two. When the register goes to zero, the DMA GO bit in the DMCSR is cleared thereby stopping the DMA logic. The RDY bit in the DMCSR is set causing an interrupt to the T11. Figure 2-5 shows the bit format of the DMWC register.



TK-9801

Figure 2-5 DMWC Register Bit Format and Bit Descriptions

2.2.1.2 T11 UNIBUS DMA -- The T11 UNIBUS DMA is used by the port microprocessor to access the host memory in order to perform the following functions:

- Read ring structure data,
- Read data buffers from host memory for transmission by the link, and
- Write status information to data buffers upon completion of transmission.

A T11 UNIBUS DMA transactions occurs in the following sequence:

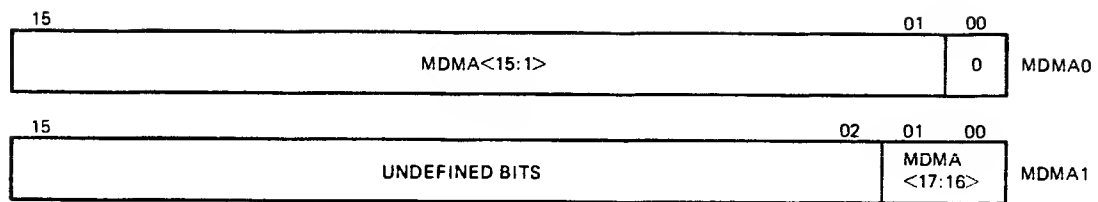
1. The T11 loads the UNIBUS address registers, MDMA0 and MDMA1.
2. The T11 either reads or writes the microprocessor DMA data register. (MDMDR0 incrementing or MDMDR1 decrementing. Refer to Section 2.2.3.1.)
3. The reading or writing of the data register causes the microprocessor DMA to acquire the UNIBUS and transfer the data to/from host memory. During the DMA process, the T11 is stalled until the transfer is complete.

2.2.1.2.1 Microprocessor DMA Address Registers (MDMA0 and MDMA1)
-- The microprocessor DMA address registers are made up of a 17-bit counter that contains the address in host memory to or from which the data is to be transferred. MDMA0 contains the lower 16 bits of the address and MDMA1 contains the upper 2 bits of the address.

NOTE

Bit 0 of MDMA0 is always a 0 because the
DMA logic only performs word transfers.

Figure 2-6 shows the Microprocessor DMA Address Register Bit Format.



TK-8802

Figure 2-6 Microprocessor DMA Address Register Bit Format and Description

2.2.1.2.2 Microprocessor DMA Data Registers (MDMDR0 and MDMDR1)
 -- The microprocessor DMA data registers are used as data ports for the data that is transferred to/from host memory. If the T11 reads/writes the first register, MDMDR0, the address contained in MDMA0 and MDMA1 is incremented by two. If the T11 reads/writes the second register, MDMDR1, the address contained in MDMA0 and MDMA1 is decremented by two. This allows the T11 to do multiple transfers without loading the host memory address for each transfer. The reading or writing of MDMDR0 or MDMDR1 by the T11 generates an NPR request to the UNIBUS. Refer to Figures 2-7 and 2-8 for MDMDR0 and MDMDR1 bit formats.

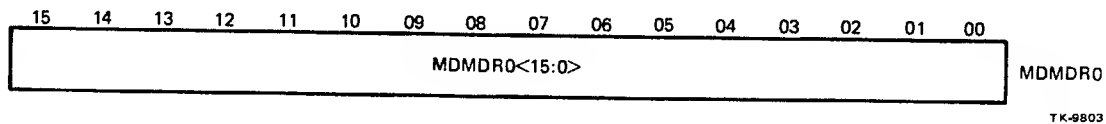


Figure 2-7 Data Register MDMDR0 (Incrementing)

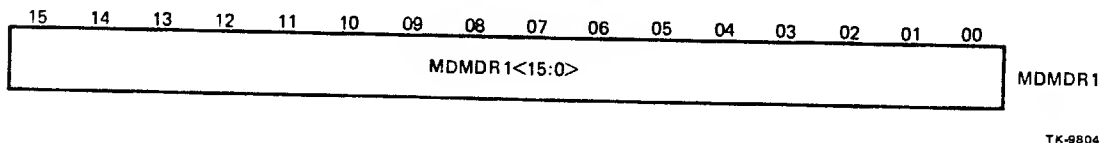


Figure 2-8 Microprocessor DMA Data Register MDMDR1

2.2.2 Port Control and Status Registers

The port control and status registers (PCSR) are used by the port module to receive commands from the host processor and report the results of the command along with other status information (interrupts, etc.).

There are four PCSRs, each with a specific function. The following sections show the format of the PCSRs and give a description of their function.

For a more detailed explanation of functions performed by the PCSRs, refer to Chapter 4, "Programming", of the DEUNA User's Guide (EK-DEUNA-UG).

2.2.2.1 Port Control and Status Register 0 (PCSR0) -- Figure 2-9 shows the format of PCSR0 and Table 2-2 lists the functions of the bits.

15	14	13	12	11	10	09	08	07	06	05	04	03	00	
SERI	PCEI	RXI	TXI	DNI	RCBI	0	USCI	INTR	INTE	RSET	0	PORT_COMMAND	PCSR0	
RWCL	RWCL	RWCL	RWCL	RWCL	RWCL	0	RWCL	R	R/W	W	0	R/W	PORT DRIVER ACCESS	
W	W	W	W	W	W	0	W	W	R	R	0	R	PORT ACCESS	
0	0	0	0	0	0	0	0	0	0	0	0	U	POWER UP STATE	

TERMS

RWCL	READ ACCESS, WRITE ONE TO CLEAR
R/CL	READ ACCESS, CLEAR
R	READ ONLY, IGNORED WHEN WRITTEN
R/W	READ/WRITE
W	WRITE ONLY, READ AS ZERO
U	UNDEFINED

TK-0068

Figure 2-9 PCSR0 Format

Table 2-2 PCSR0 Bit Descriptions

Bits	Name	Description
<15>	SERI	Status Error Interrupt -- Indicates the presence of an error condition flagged in status register accessible by the port command function. Set by the DEUNA, cleared by the port-driver through the read and clear status port function.
<14>	PCEI	Port Command Error Interrupt -- Indicates the occurrence of either a function error or a UNIBUS timeout during the execution of a port command. Bit 7 of PCSR1 distinguishes between the two error conditions. Set by the DEUNA, cleared by the port-driver.
<13>	RXI	Receive Ring Interrupt -- Attention bit for ring updates. Set the by the DEUNA cleared by the port-driver. Indicates, when set, that the DEUNA has placed a message(s) on the ring.
<12>	TXI	Transmit Ring Interrupt -- Attention bit for ring updates. Set by the DEUNA, cleared by the port-driver. Indicates, when set, that transmission has been suspended. All messages it found on the transmit ring have been set, or an error was encountered during a transmission.
<11>	DNI	Done Interrupt -- Interrupts when the DEUNA completes a port command. (Note: the port command NO-OP does not cause the DNI bit to set.) Set by the DEUNA, cleared by the port-driver.
<10>	RCBI	Receive Buffer Unavailable Interrupt -- Interrupts when the DEUNA discards an incoming message due to receive ring buffers being unavailable. Once set by the DEUNA, RCBI is not set again until after the DEUNA has received a PDMD port command and has discarded a subsequent message. Set by the DEUNA, cleared by the port-driver.
<09>	zero	

Table 2-2 PCSR0 Bit Descriptions (Cont)

Bits	Name	Description
<08>	USCI	<p>Unsolicited State Change Interrupt -- Interrupts when the DEUNA performs the following actions:</p> <p>Fatal Error -- A transition into the NI and UNIBUS halted state from the ready, running, UNIBUS halted, or NI halted states. This state change is caused by the DEUNA detecting an internal fatal error, that is, internal parity error.</p> <p>Communication Processor Boot -- A transition into the primary load state caused by the reception of a remote boot request of the communication processor (DEUNA microcode).</p> <p>Communication Processor Boot -- A transition into the ready state from the primary load state following the reception of the memory load with transfer address message, as part of a remote boot request.</p> <p>The three conditions are distinguished by examining the state field of PCSR1. Set by the DEUNA, cleared by the port-driver.</p>
<07>	INTR	Interrupt Summary -- The logical OR of PCSR0 <15:08>. Set by the DEUNA.
<06>	INTE	Interrupt Enable -- Set or cleared by the port-driver, unchanged by the DEUNA.

NOTE

In order to overcome synchronization problems with the port command field when writing the INTE bit, the DEUNA hardware locks the port command field during write accesses that change the INTE bit from a one to a zero or change the INTE bit from a zero to a one. Issuing the DEUNA, a port command, and changing the state of the INTE bit must occur in two different write accesses.

Table 2-2 PCSR0 Bit Descriptions (Cont)

Bits	Name	Description
<05>	RSET	DEUNA Reset -- Clears the DEUNA and returns it to the power up state when written with a one byte port-driver. This bit is write-only. After a successful reset, PCSR0 <11> (DNI) = 1 and PCSR0 <07> (INTR) = 1.
<04>	zero	
<03:00>	PORT_COMMAND	
	0 0 0 0	NO-OP No operation
	0 0 0 1	GET PCBB Instructs the DEUNA to fetch the address of the port control block from PCSRs 2 and 3. The DEUNA accesses PCSRs over the UNIBUS conductor, and retains a copy of the address internally. If the address of the port control block is changed, this command must be repeated to inform the DEUNA.
	0 0 1 0	GET CMD Instructs the DEUNA to fetch and execute a command found in the first word of the port control block. The address of the port control block was obtained through the get PCBB command.
	0 0 1 1	SELF TEST Instructs the DEUNA to enter the reset state and execute self-test.
	0 1 0 0	START Enables transmission and reception of frames from the port-driver. This command is ignored by the DEUNA if it is in the running state. Clears any current buffer status that

Table 2-2 PCSR0 Bit Descriptions (Cont)

Bits	Name	Description
		the DEUNA has stored internally and resets the ring pointers to the base addresses of the rings.
0 1 0 1	BOOT	Instructs the DEUNA to enter the primary load state and initiate the down-line load of additional DEUNA microcode.
0 1 1 0	Not Used	Reserved code, causes a NO-OP.
0 1 1 1	Not Used	Reserved code, causes a NO-OP.
1 0 0 0	PDMD	Polling Demand -- Instructs the DEUNA to check the descriptor rings. The DEUNA polls the receive descriptor ring only if it had not previously acquired a free buffer.
1 0 0 1	Not Used	Reserved code, causes a NO-OP, sets DNI.
1 0 1 0	Not Used	Reserved code, causes a NO-OP, sets DNI.
1 0 1 1	Not Used	Reserved code, causes a NO-OP, sets DNI.
1 1 0 0	Not used	Reserved code, causes a NO-OP, sets DNI.
1 1 0 1	Not Used	Reserved code, causes a NO-OP, sets DNI.
1 1 1 0	Not Used	Reserved code, causes a NO-OP, sets DNI.
1 1 1 1	STOP	Suspends the operation of the DEUNA to transition to the ready state. Causes no action if the DEUNA is not in the running state.

2.2.2.2 Port Control and Status Register 1 (PCSR1) -- Figure 2-10 shows the format of PCSR1 and Table 2-3 lists the functions of the bits.

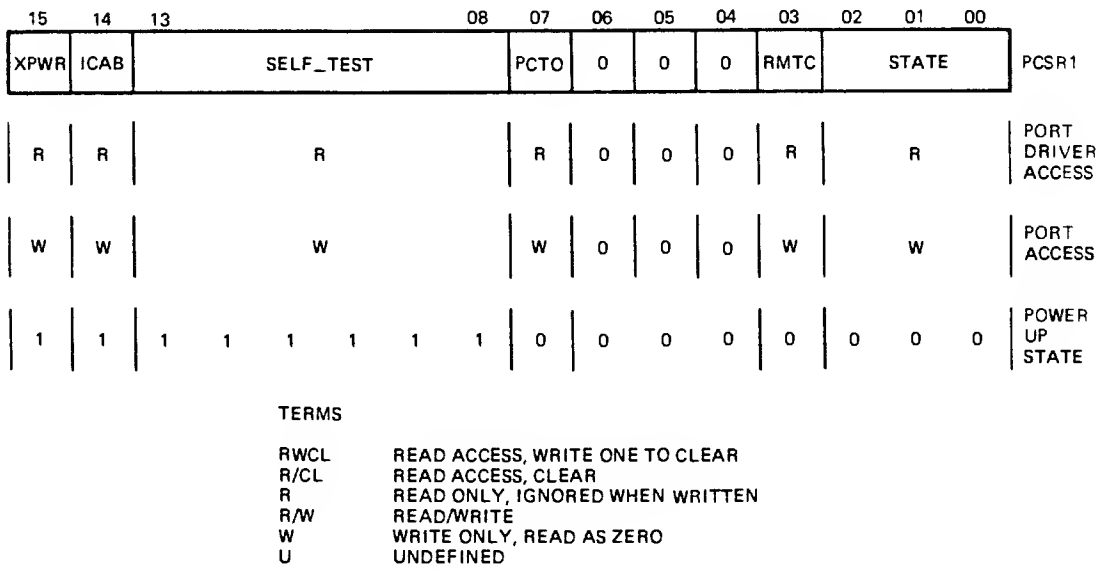
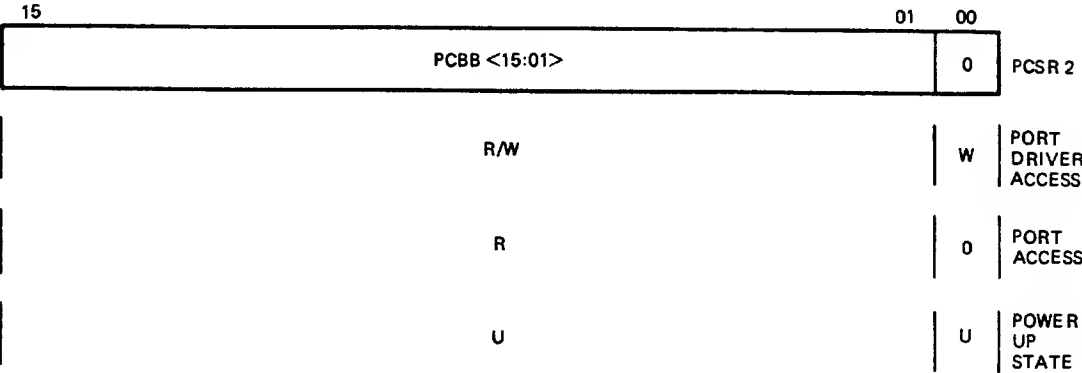


Figure 2-10 PCSR1 Format

Table 2-3 PCSR1 Bit Descriptions

Bits	Name	Description																				
<15>	XPWR	Transceiver Power OK -- A zero indicates that a failure exists in either the transceiver power supply or the fuse on the link module.																				
<14>	ICAB	Port/Link Cabling OK -- A zero indicates that the interconnecting cable between the port and link modules has a seating problem.																				
<13:08>	SELF-TEST	Self-Test Error Code -- The encoded test of the DEUNA failed during self-test. A code of zero indicates no failure.																				
<07>	PCTO	Port Command Timeout -- A UNIBUS timeout was encountered while executing a port command. Valid only after the PCEI bit of PCSR0 is set by the DEUNA. This bit is used to distinguish between a DEUNA failure to complete a port command due to a UNIBUS timeout or a function error.																				
<06:04>	Zeros																					
<03:00>	STATE	<table><tr><td>0 0 0 0</td><td>Reset</td></tr><tr><td>0 0 0 1</td><td>Primary Load</td></tr><tr><td>0 0 1 0</td><td>Ready</td></tr><tr><td>0 0 1 1</td><td>Running</td></tr><tr><td>0 1 0 0</td><td>Not Used</td></tr><tr><td>0 1 0 1</td><td>UNIBUS Halted</td></tr><tr><td>0 1 1 0</td><td>NI Halted</td></tr><tr><td>0 1 1 1</td><td>NI and UNIBUS Halted</td></tr><tr><td></td><td>Fatal internal error, that is parity error. An interrupt condition. When the DEUNA is in this state, the FATI bit of PCSR0 is also set. Cleared by the port-driver setting the RSET bit.</td></tr><tr><td>1 1 1 1</td><td>Secondary Loader</td></tr></table>	0 0 0 0	Reset	0 0 0 1	Primary Load	0 0 1 0	Ready	0 0 1 1	Running	0 1 0 0	Not Used	0 1 0 1	UNIBUS Halted	0 1 1 0	NI Halted	0 1 1 1	NI and UNIBUS Halted		Fatal internal error, that is parity error. An interrupt condition. When the DEUNA is in this state, the FATI bit of PCSR0 is also set. Cleared by the port-driver setting the RSET bit.	1 1 1 1	Secondary Loader
0 0 0 0	Reset																					
0 0 0 1	Primary Load																					
0 0 1 0	Ready																					
0 0 1 1	Running																					
0 1 0 0	Not Used																					
0 1 0 1	UNIBUS Halted																					
0 1 1 0	NI Halted																					
0 1 1 1	NI and UNIBUS Halted																					
	Fatal internal error, that is parity error. An interrupt condition. When the DEUNA is in this state, the FATI bit of PCSR0 is also set. Cleared by the port-driver setting the RSET bit.																					
1 1 1 1	Secondary Loader																					

2.2.2.3 Port Control and Status Register 2 (PCSR2) -- Figure 2-11 shows the format of PCSR2 and Table 2-4 lists the functions of the bits.



TERMS	
RWCL	READ ACCESS, WRITE ONE TO CLEAR
R	READ ONLY, IGNORED WHEN WRITTEN
R/W	READ/WRITE
W	WRITE ONLY, READ AS ZERO
U	UNDEFINED

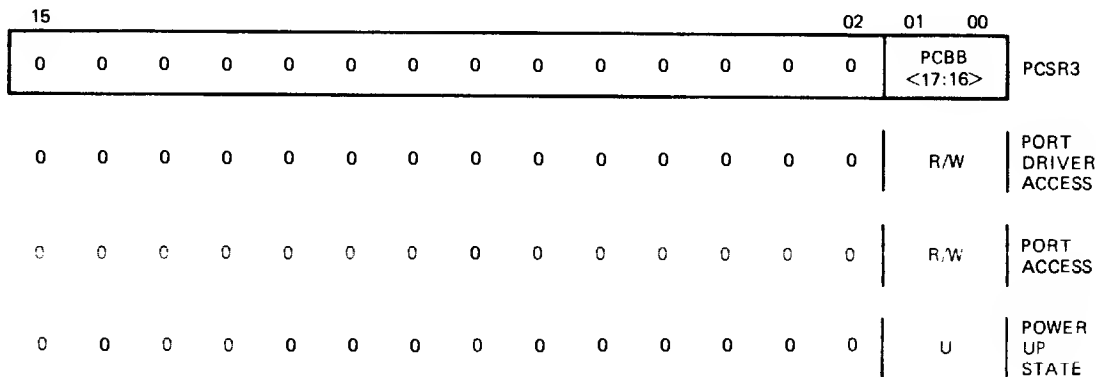
TK-0070

Figure 2-11 PCSR2 Format

Table 2-4 PCSR2 Bit Description

Bits	Name	Description
<15:00>	PCBB	The low order 16 bits of the address of the port control block base. The PCBB is read by the port as an even number.

2.2.2.4 Port Control and Status Register 3 (PCSR3) -- Figure 2-12 shows the format of PCSR3 and Table 2-5 lists the function of the bits.



TERMS

RWCL READ ACCESS, WRITE ONE TO CLEAR
R READ ONLY, IGNORED WHEN WRITTEN
R/W READ/WRITE
W WRITE ONLY, READ AS ZERO
U UNDEFINED

TK-0071

Figure 2-12 PCSR3 Format

Table 2-5 PCSR3 Bit Description

Bits	Name	Description
<15:02>		Zeros
<01:00>	PCBB	The high order 2 bits of the address of the port control block base.

2.3 MICROPROCESSOR SECTION

The function of the microprocessor section of the port module is to:

- Manage the ring structure in host memory,
- Set up the DMA control for the transfer of data between host memory and the link module (receive and transmit functions), and
- Interpret received or transmitted packets.

The microprocessor section consists of the following components:

1. T11 microprocessor,
2. DAL/BDAL-time multiplexed data/address bus,
3. T11 address latch,
4. 8K words of PROM storage-microcode,
5. 4K words of RAM storage-writable control store (WCS), and
6. Internal I/O decode-used when T11 has to access a register on the PORT module.

2.3.1 Microprocessor

The DEUNA uses a microprocessor located on the port module to control its operation. The microprocessor used is a DCT11-AA (T11). The T11 is a single chip microprocessor that uses the LSI-11 instruction set. The T11 communicates to the port module over a time multiplexed bidirectional bus called the data address lines (DAL). It also receives process and status information via a separate set of interrupt inputs. Each interrupt and its function is listed in Table 2-6.

The T11 can access a total of 32K words of memory. This address space is divided into areas for:

- Microprogram storage,
- Writable control store (WCS),

- Transmit and receive buffer space, and
- Input/output control.

Figure 2-13 shows the configuration of the T11's address space.

For more information on the operation of the T11 microprocessor, refer to the DCT11-AA Microprocessor User's Guide (EK-DCT11-UG).

Table 2-6 T11 Interrupts

Interrupt	Signal Name	Description
Receive Miss	MISS INTR	There is no receive buffer available for an incoming message.
Memory Parity	LNK MEM PAR ERR	There is a parity error in the link buffer memory.
PCSR Write	PCSR INTR	The host processor has written a command into PCSR0.
UNIBUS Error	UBERR INTR	There is a UNIBUS timeout.
Transmit Done	XMIT DONE	The link has finished transmitting a buffer.
Receive Buffer Done	RCV BUFF DONE	There is a receive buffer waiting to be sent to host memory.
Timer	TIMER INTR	Interrupts T11 every second for timing information.
DMA Ready	DM INTR	DMA machine ready to be started.

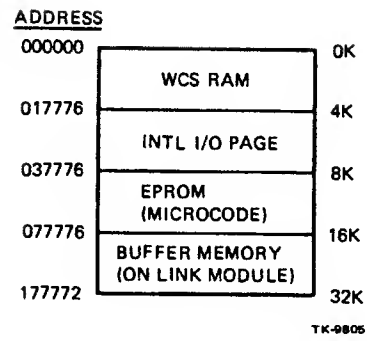


Figure 2-13 T11 Address Space

2.3.2 Internal Registers

The internal registers of the port module are used by the T11 for setting up and controlling the operation of the DEUNA. These registers reside in the I/O page of the T11.

The I/O decode logic of the port enables the selected internal register when it is addressed by the T11. This logic monitors the output of the T11 address latch and the read and write signals generated by the T11.

Table 2-7 gives a list of the internal register addresses and the type of access allowed.

Table 2-7 Internal Register Address Assignments

Address	Name	Access	Description
21000	PCSR0	R/W	Port control and status reg. 0
21002	DMCSR	R/W	DMA control and status reg.
21004	DMAT0	R/W	DMA-to-address register 0
21006	DMAT1	R/W	DMA-to-address register 1
21010	MDMA0	R/W	MicroCPU DMA-to-adrs. reg. 0
21012	MDMA1	R/W	MicroCPU DMA-to-adrs. reg. 1
21014	MDMDR0	R	MicroCPU DMA data reg. 0
21016	MDMDR1	R	MicroCPU DMA data reg. 1
21020	PCSR1	WO	Port control and status reg. 1
21022	DMAF	WO	DMA-from-address register
21024	DMWC	WO	DMA word count register
21026	MDMDR0	WO	Read Inc UB data port
21030	LTAC	WO	Link transmit adrs. counter reg.
21032	LRBAF	WO	Link rec. buffer address FIFO
21034	LCSR	WO	Link control and status reg.
21036	MDMDR1	WO	Write Dec. UB data port
21040	PCSRSW	RO	Port switchpack reg.
21042	UNUSED	RO	

Table 2-7 Internal Register Address Assignments (Cont)

Address	Name	Access	Description
21044	LCBAF	RO	Link completed buffer add. FIFO
21046	PCSR1	RO	Port cntl. and status reg. 1
21050	UNUSED	RO	
21052	UNUSED	RO	
21054	UNUSED	RO	
21056	UNUSED	RO	
21060	PHYAD0	RO	Physical address byte 0
21062	PHYAD1	RO	Physical address byte 1
21064	PHYAD2	RO	Physical address byte 2
21066	PHYAD3	RO	Physical address byte 3
21070	PHYAD4	RO	Physical address byte 4
21072	PHYAD5	RO	Physical address byte 5
21074	PHYAD6	RO	Physical address byte 6
21076	PHYAD7	RO	Physical address byte 7

2.3.3 Default Station Address (Physical Address)

The microprocessor section of the DEUNA contains a PROM which the T11 can read on power up to get the default address of the node. When the T11 reads the physical address from the PROM it transfers it to the station address RAM on the link module.

The physical address in the station address RAM can be changed by the host by a change physical address command.

2.3.4 Physical Address Registers

These registers are used to read the physical address from the physical address PROM. Figure 2-14 shows the configuration of these registers.

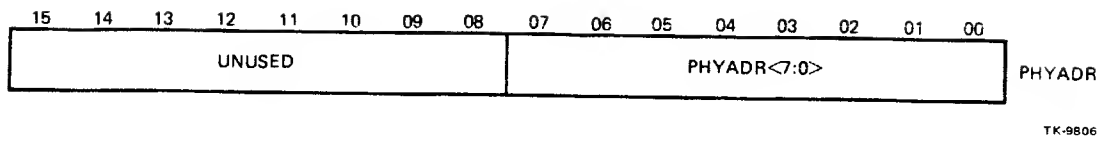
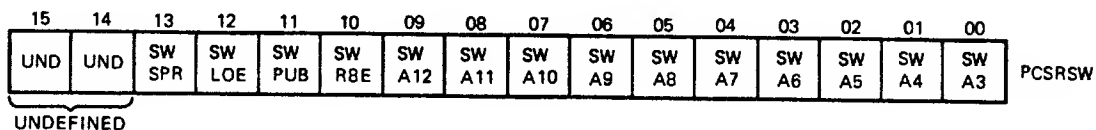


Figure 2-14 Physical Address Register Bit Configuration

2.3.5 Port Switchpack Register

This register allows the microprocessor to read the switch selected UNIBUS address of the PCSRs and the function switches on the port module. Figure 2-15 shows the configuration of the register and Table 2-8 gives a description of the bits.



TK-9807

Figure 2-15 Port Switchpack Register Bit Configuration

Table 2-8 Port Switchpack Register Bit Descriptions

Bit	Field	Description
PCSRSW<09:00>	SW AXX	UNIBUS address address
PCSRSW<10>	SW RBE	Remote boot enable switch
PCSRSW<11>	SW PUB	Power-up boot switch
PCSRSW<12>	SW LOE	Loop-on-self-test error switch
PCSRSW<13>	SW SPR	Spare switch
PCSRSW<15:14>	UND	Undefined

2.3.6 Timer

The timer is made up of a one shot that generates an interrupt to the T11 every second. This allows the T11 to time events through the use of software routines.

2.3.7 Internal Buses

The port uses three sets of internal buses for the transfer of information within the DEUNA. These buses are:

1. DAL/BDAL (Data/Address Lines, Buffered Data/Address Lines).
 - Time multiplexed -- carry data during part of the timing cycle and address during the other part of the timing cycle.
 - BDAL is a buffered extension of the DAL for loading purposes.
2. T/F BUS (To/From Bus) -- transfers data between the UNIBUS bus and the DEUNA.
3. LMD BUS (Link Memory Data Bus) -- data bus to link memory buffers.
4. LINK MEM A (Link Memory Address Bus) -- address bus to link memory buffers.

2.4 LINK MEMORY CONTROL

The link memory section is the part of the port module which communicates with the link module. This section contains control for the 16K words of RAM that are located on the link module (parity generation and memory are on the port module). This memory is divided into 16 buffers that are used to buffer packets of data being transmitted to or received from the ETHERNET via the link module.

The link memory section contains the logic necessary to:

1. Arbitrate for use of the link memory,
2. Keep track of which buffers are available for use, and
3. Generate the memory addresses for writing or reading data from memory.

2.4.1 Link Memory Arbitration

Link memory is accessed by four different processes:

1. Link transmit state machine,
2. Link receive state machine,
3. DMA control, and
4. T11.

Arbitration for use of link memory by any of these processes is performed by the link memory arbitration PAL. A description for the PAL is given in the DEUNA engineering drawings.

2.4.2 Link Transmit Address Counter (LTAC)

The link transmit address counter is used when a transmit buffer is to be transmitted onto the ETHERNET.

The link transmit address counter consists of two sections:

1. Link Transmit Address Counter Register -- loaded by T11 with the four-bit buffer address.
2. Link Transmit Address Counter -- this is a 10-bit counter that is used to generate the lower 10 bits of the transmit buffer address.

They are configured as shown in Figure 2-16.

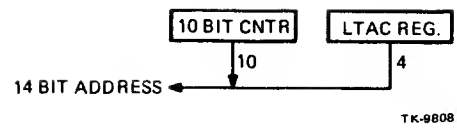


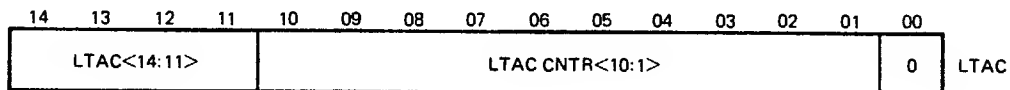
Figure 2-16 LTAC Configuration

When a transmit buffer in link memory is to be transmitted onto the ETHERNET, the following action takes place.

1. The T11 loads the LTAC register with the four-bit buffer address. This clears the 10-bit counter and notifies the transmit state machine on the link module that there is a buffer to be transmitted.
2. Transmit state machine increments counter by two after reading the word to be transmitted until the buffer is empty.

The transmit state machine can clear the 10-bit counter if it needs to do a transmit retry.

Figure 2-17 shows the bit configuration of the LTAC.



TK-9809

Figure 2-17 LTAC Bit Configuration

2.4.3 Link Receive Address Counter

The link receive address counter is used to generate the buffer addresses for messages received from the ETHERNET by the DEUNA.

The link receive address counter is made up of three sections.

1. The link receive buffer address FIFO (LRBAF),
2. The link completed buffer address FIFO (LCBAF), and
3. The link receive address counter.

They are configured as shown in Figure 2-18.

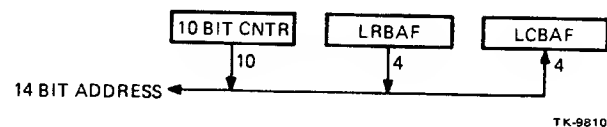
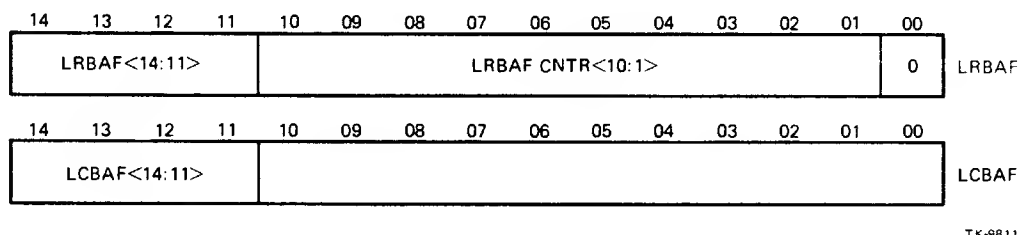


Figure 2-18 Receive Address Counter Configuration

The LRBAF and the LCBAF are four-bit by 64 location FIFOs. The LRBAF has the upper four bits of all available buffer addresses placed into it by the T11. When a receive buffer is needed by the link, the following functions are performed.

1. The address counter is cleared.
2. The buffer address at the output of the LRBAF and the output of the counter are used to generate the link memory address.
3. The counter is incremented until the buffer is completed.
4. When the buffer is completed by the link it advances the LRBAF which loads the address of the completed buffer into the LCBAF and clears the counter. The address bubbles through the FIFO.
5. When a buffer address is available at the output of the LCBAF the T11 is notified that there is a completed receive buffer. This is done by generating an interrupt (RCV BUF DONE) to the T11.
6. The T11 then processes the completed buffer and returns the buffer address to the LRBAF.

Figure 2-10 shows the configuration of the LRBAF and the LCBAF.



TK-9811

Figure 2-19 LRBAF, LCBAF Bit Configuration

2.4.4 T11 Addressing of Link Buffer Memory

When the T11 addresses link buffer memory, the memory arbiter arbitrates for use of the memory. When the T11 receives use of the memory the requested data is passed to the T11. During the data transfer the T11 is stalled.

2.4.5 Port-to-Link Interface

The DEUNA is comprised of two modules which have to be UNIBUS SPC compatible. This does not allow for a backplane interconnect. Therefore, the DEUNA port and link modules are connected by a Berg type connector over the handles cables. The signals on these cables comprise the port link interconnect.

The signals on these cables are broken down into five classes:

1. Link memory bus signals,
2. Link memory address control signals,
3. Link command register signals,
4. Link discrete status signals, and
5. Clock and initialize signals.

The following sections describe the port-to-link interface signals.

2.4.5.1 Link Memory Bus --

Signal	Source	Description
BUS LMD <15:00>	BIDIR	Link Data Bus -- Sixteen bidirectional data lines between the link and port modules.
LINK MEM A <14:01>	PORT	Link Memory Address Bus -- Fourteen address lines between the port and link modules.
BUS READ	BIDIR	Read/Write -- Used to indicate the direction of the transfer.
RX REQUEST	LINK	Receiver Request -- Used by the link receive state machine to request the link memory.
RX ACK	PORT	Receiver Acknowledge -- Used by the port to acknowledge the link request.
TX REQUEST	LINK	Transmit Request -- Used by the link transmit state machine to request the link memory.

TX ACK	PORT	Transmit Acknowledge -- Used by the port to acknowledge the link request.
--------	------	---

2.4.5.2 Link Memory Address Control Signals -- A detailed description of these signals is given in Chapter 3 of this manual.

Signal	Source	Definition
INC TX POINTER	LINK	Increment Transmit Pointer -- Used by the link to increment the transmit address pointer.
RES TX POINTER	LINK	Restore Transmit Pointer -- Used by the link to restore the transmit address counter to the beginning address.
INC RX POINTER	LINK	Increment Receiver Pointer -- Used by the link to increment the receive address counter.
RES RX POINTER	LINK	Restore Receiver Pointer -- Used by the link to restore the receive pointer to the beginning.
ADV RX POINTER	LINK	Advance Receiver Pointer -- Used by the link to get the next receive address buffer.

2.4.5.3 Command Register Control --

Signal	Source	Description
CMDW	PORT	Command Register Write -- Enables the command register to be written from the link memory bus.
CMDE	PORT	Command Register Execute -- Tells the link to execute the command in the link command register.

2.4.5.4 Link Discrete Status --

Signal	Source	Description
CERR	LINK	Collision Test Error -- Indicates the collision output failed to activate during the collision test following a transmission (heartbeat).

SET MISS	LINK	Missed Packet -- Indicates that the link failed to write a received packet into link memory because a buffer was unavailable.
TATT	PORT	Transmitter Attention -- Tells the link that the port has completed a buffer for transmission.
TX DONE	LINK	Transmit Done -- Indication to the port that the link has finished transmitting a buffer.
ICAB1	LINK	Installed Cable 1 & 2 -- Used by the port to re-ensure the interconnecting cable is plugged in properly.
ICAB2	LINK	
FUSE CHECK	LINK	Transceiver Power -- Used by the port to check that the power to the transceiver is available.
RX FREE BUF	PORT	Receiver Buffer Free -- Used by the link to find out if there are any free receive buffers.
WR RESET	PORT	Reset -- Used by the link to do a UNIBUS reset.

2.4.5.5 Clock and Reset --

Signal	Source	Description
10MHZ	LINK	Clock -- 10 MHz square wave.
INIT	PORT	Buffered Initialize -- Buffered UNIBUS INIT.

CHAPTER 3
LINK MODULE
FUNCTIONAL DESCRIPTION

3.1 INTRODUCTION

The link module (M7793) is the interface between the DEUNA and the ETHERNET transceiver. It is microprogram controlled and provides the following functions.

- Physical channel interface
- Parallel-to-serial conversion of data on transmit
- Serial-to-parallel conversion of data on receive
- Collision detection and retry
- CRC generation and checking
- Station address detection
- Link memory bus control

The link in connection with the port provides the logic necessary to interface the UNIBUS Bus with the ETHERNET.

A functional block diagram of the link module is shown in Figure 3-1. The letters, in parenthesis, on the block diagram give the location of the logic for that functional block in the engineering drawings.

3.2 LINK MEMORY BUS

The link memory bus provides the communication path between the link module and the port module. The bus is made up of 54 lines that are divided into four signal groups.

- Memory Bus
- Discrete Control
- Discrete Status
- Clock

Tables 3-1, 3-2, 3-3, and 3-4 list the link memory bus signal names, their source, and a description of their function.

Table 3-1 Memory Bus Signals

Signal	Source	Description
LINK MEM <14:01>	Port	Link Memory Address Bus - Fourteen address lines used to address the memory buffers on the link module.
BUS LMD <15:00>	BIDIR	Link Memory Data Bus - Sixteen bidirectional data lines between the link and port module.
TX REQUEST	Link	Transmit Data Request - Used by the link to start arbitration for the Link Memory Bus.
RX REQUEST	Link	Receive Data Request - Used by the link to start arbitration for the link memory data bus.
TX ACK	Port	Transmit Data Acknowledge - Used by the port to inform the link that it has granted the link memory bus for a transmit operation.
RX ACK	Port	Receive Data Acknowledge - Used by the port to inform the link that it has granted the link memory bus receive operation.
BUS READ	BIDIR	Read/Write - Used to indicate the direction of the transfer. When set data is transferred from a link memory buffer.

Table 3-2 Discrete Control Bus Signals

Signal	Source	Description
INIT	Link	Synchronized Initialize - Clock synchronized power up initialize.
WR RESET	Port	Software Reset - Comes from port PCSR0.
CMDW	Port	Command Register Write - Enables the command register on the link to be written from the link memory bus. This signal is valid for 100 ns.
TATT	Port	Transmitter Attention - The port notifies the link that a transmit buffer is ready for transmission. Set by the port cleared by TX DONE.

Table 3-2 Discrete Control Bus Signals (Cont)

Signal	Source	Description
RES TX POINTER	Link	Reset Transmit Pointer - Tells the port to reset the transmit address pointer on the port. This signal is valid for 100 ns.
INC TX POINTER	Link	Increment Transmit Pointer - Tells the port to increment the transmit address pointer on the port. This signal is valid for 100 ns.
RES RX POINTER	Link	Reset Receive Pointer - Tells the port to reset the receiver address pointer on the port. This signal is valid for 100 ns.
INC RX POINTER	Link	Increment Receiver Pointer - Tells the port to increment the receiver address pointer on the port. This signal is valid for 100 ns.
ADV RX POINTER	Link	Advance Receiver Pointer - Tells the port to advance the receive buffer address pointer on the port. This signal is valid for 100 ns.
CABLE VERIFY IN	Port	Cable Verify Input - This circuit provides a closed loop electrical path with cable verify output that is used to indicate that the cable between the link and the port is installed and connected properly.
CABLE VERIFY OUT	Port	Cable Verify Output

Table 3-3 Discrete Status Bus Signals

Signal	Source	Description
CERR	Link	Collision Test Error - The transceiver collision output failed to activate during the collision test following transmission (heartbeat). Set during a collision test error. This signal is valid for 100 ns. This signal is valid for the H4000 or equivalent transceiver.
MISS	Link	Missed Packet - Receiver failed to write a packet addressed to the port into the link memory because a buffer was unavailable. This signal is valid for 100 ns.

Table 3-3 Discrete Status Bus Signals (Cont)

Signal	Source	Description
TX DONE	Link	Transmit Done - Indication to the port that the link has finished transmitting a buffer. This signal is valid for 100 ns.
FUSE CHECK	Link	Transceiver Power OK - A ONE indicates that a failure exists in either the transceiver power supply or in the cabling to the bulk-head assembly.
FREE RX BUFF	Port	Free Receiver Buffer - A buffer is available in the link memory to put an incoming packet. Set by the port, cleared by ADV RX pointer.

Table 3-4 Clock Signal

Signal	Source	Description
10 MHz	Link	Clock - The link clock is a 100 nanosecond square wave derived from a free running 10 MHz clock located in the ECL section of the link.

3.3 LINK REGISTERS

The operation of the link module is controlled by the port module though the use of two registers. The two registers are the link command register and the mode register. These registers are used to initialize, start, stop, and select the mode of operation of the link module. In addition to the command and mode registers, the link contains the station address RAM. The station address RAM is used to hold the addresses of the node for decoding by the address detection logic.

3.3.1 Command Register

The link command register is used by the port module to initialize, start, and stop the link module. This register is accessed by the port microprocessor by asserting CMDW H on the link memory bus. This register is write only by the port and is set to all zeros on power up or when initialized.

Figure 3-2 shows the format of the register and Table 3-5 describes the function of each bit.

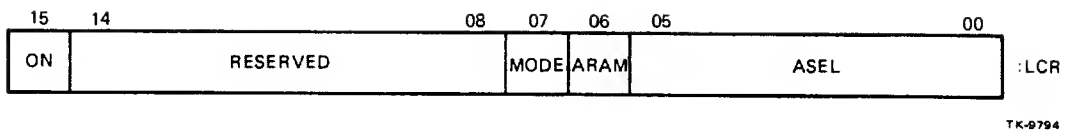


Figure 3-2 Format of Link Command Register

Table 3-5 Link Command Register Bit Descriptions

Bits	Field	Description
<15>	ON	Enable Link Module - When set, this bit enables both the receive and the transmit state machines. Set and cleared by the port. Powers on in the zero state.
<14:8>	Reserved	
<7>	Mode	Enable Mode Register - When set, this bit enables the write access of the mode register over the link memory data bus when CMDE H is asserted by the port. Set and cleared by the port.
<6>	ARAM	Enable Station Address RAM - When set, this bit allows the station address RAM to be written when CMDE H is asserted by the port. Set and cleared by the port.
<5:0>	ASEL	Address Select - Specifies the memory location within the station address RAM containing the physical and logical address: the data section of the station address begins at location ASEL=20 (octal). Set and cleared by the port.

NOTE

The first word, ARAM0, of the Station Address RAM begins at location ASEL=20 (octal). This is due to the binary counter logic used in the address comparator section.

3.3.2 Link Mode Register

The port uses the mode register to control the transmit and receive operations of the link module. It is written when the mode bit of the link command register is set and bus signal CMDE is asserted. The register is set to all zeros on power up or when the link is initialized.

Figure 3-2 shows the format of the register and Table 3-6 describes the function of each bit.

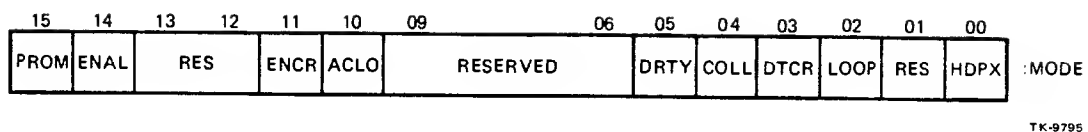


Figure 3-3 Link Mode Register Format

Table 3-6 Bit Descriptions for Link Mode Register

Bits	Field	Description
<15>	PROM	Instructs the link to accept all incoming frames regardless of the destination address field. Written and cleared by the port.
<14>	ENAL	Instructs the link to accept all incoming frames with multicast destinations. Written and cleared by the port.
<13:12>	RES	Reserved
<11>	ENCR	Enable Collision Test Error. When set, any collision test errors will be reported back to the port. Set and cleared by the port.
<10>	ACLO	Enable ACLO. When set, ACLO asserts ACLO on the UNIBUS Bus and disables INIT on the DEUNA. Set by the port cleared by the link.
<9:6>	RESERVED	
<5>	DRTY	Disable Retry Logic. When set, the link attempts only one transmission of a packet. This is a maintenance self-test function. Written and cleared by the port.
<4>	COLL	Simulate a collision on the wire during loopback mode. This is a maintenance self-test function. Written and cleared by the port.
<3>	DTCR	Disable Transmit CRC Logic. If DTCR=1, the CRC logic is dedicated to the receiver. If DTCR=0, the CRC logic is dedicated to the transmitter. This feature is used as a loopback maintenance function. Written and cleared by the port.
<2>	LOOP	Enable Loopback. When set, this bit enables loopback internal to the link, and the CRC logic dedicated to the receiver or transmitter as selected by DTCR. Written and cleared by the port.
<1>	RES	Reserved

Table 3-6 Bit Descriptions for Link Mode Register (Cont)

Bits	Field	Description
<0>	HDPX	Half-Duplex Mode. Indicates when clear that the link will receive messages transmitted to itself over the wire. Messages received in this manner do not undergo CRC check and a CRC error status is returned with them. Indicates when set that the link will not receive messages transmitted to itself. However, the link recognizes the transmitted messages as being addressed to itself and sets the MTCH bit in the transmit ring following the transmission attempt. Set and cleared by the port. Cleared upon power up.

3.3.3 Station Address RAM (ARAM)

The station address RAM contains the physical, logical, and broadcast addresses of the node. There can be a maximum of 12 addresses. Each address is 48 bits in length. These addresses are loaded by the port and read by the receive state machine.

Data is written to the ARAM over the link memory bus when the ARAM bit of the command register is set and the port asserts CMDE H.

Figure 3-4 shows the format of the station address RAM and Table 3-7 describes the register bits.

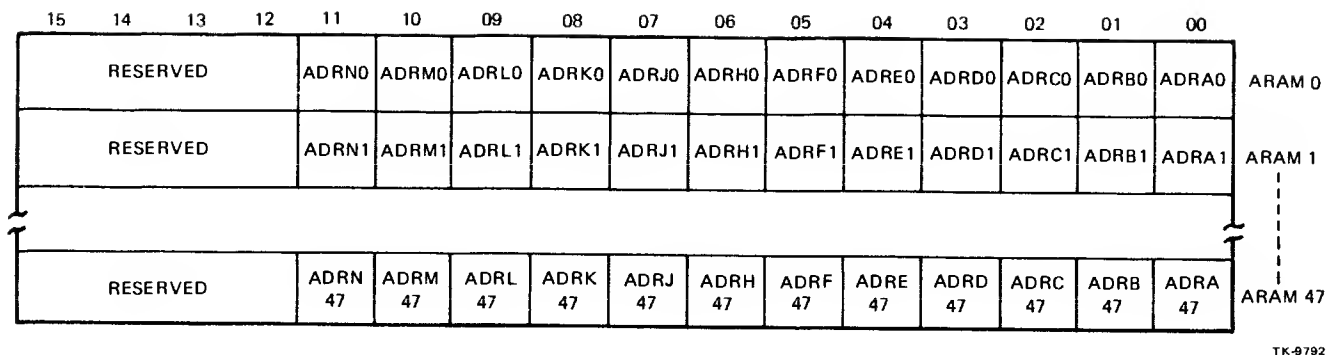


Figure 3-4 Station Address RAM Format

Table 3-7 Station Address RAM Bit Descriptions

Word	Bits	Field	Description
ARAM 0	<11:00>	ADRX0	These bits specify the first bits of each of the physical/logical/broadcast address in the station address RAM. Set and cleared by the port.
ARAM 1	<11:00>	ADRX1	These bits specify the second bits of each of the physical/logical/broadcast addresses in the station address RAM. Set and cleared by the port.
ARAM 2- ARAM 47	<11:00>	ADRX2-47	These bits specify the 2nd to 47th bits of each of the physical/logical/broadcast addresses in the station address RAM. Set and cleared by the port.

3.4 PHYSICAL CHANNEL INTERFACE

The physical channel is implemented in ECL technology and directly interfaces to the ETHERNET transceiver. The physical channel provides Manchester encoding and decoding of all serial data.

3.4.1 Transceiver Signals

The transceiver signals are those signals required by the H4000 transceiver. The following signals are the ones used to communicate between the transceiver and link.

1. Collision Presence -- This signal is used to notify the transmit and retry logic of the link of a collision on the ETHERNET.
2. Receive -- This is the data received from the ETHERNET.
3. Transmit -- This is the data to be transmitted from the link.
4. Power -- Power required for the operation of the transceiver.

3.4.2 Receiver

3.4.2.1 Receiver Squelch and Carrier Sense -- Carrier sense is asserted when one or more stations are attempting transmission on the cable, regardless of whether the station sensing carrier is transmitting at that time. Carrier sense will turn on and remain on as long as data is present on the cable.

The carrier sense signal passes through the carrier MUX and is delayed 800 ns to allow proper synchronization of the preamble.

The delayed carrier signal is used as an input to:

- CRC checker,
- Receive shifter,
- Start bit detector, and
- Receive state machine.

The nondelayed carrier signal at the output of the carrier MUX is used as an input to:

- Time Domain Reflectometer (TDR), and
- Interpacket gap counter.

3.4.2.2 Manchester Decoder -- The Manchester decoder is used to separate the incoming phase encoded bit stream from the coaxial cable into a data stream and a clock signal. The Manchester data output is used as an input to the CRC checker and the RX shifter. The RX clock generated by the Manchester decoder is used as inputs to the clock shaper, CRC checker, and the RX shifter.

3.4.2.3 Clock Shaper -- The clock shaper is used to reshape the Manchester decoder clock output to ensure a minimum clock period and pulse width. The clock shaper protects the receive clock from distortion due to noise at the receive input.

3.4.2.4 Collision Squelch -- The collision squelch is similar in operation to the receive squelch. Its output is ORed with the output of the receive squelch circuitry.

Collision is asserted when two or more stations are attempting transmission on the coaxial cable, regardless of whether the station sensing collision is transmitting at that time. The collision squelch is used as an input to the TDR counter, collision jam, and the carrier multiplexer.

This signal is synchronized to the 10 MHz system clock by a dual rank synchronizer before entering any logic operating off the system clock.

3.4.3 Transmitter

The transmitter section of the physical channel interface on the link performs the encoding of data and enables the transmitter. This logic is comprised of the Manchester encoder and transmit enable circuitry.

3.4.3.1 Manchester Encoder -- The Manchester encoder is used to translate physically separate signals of lock and data into a single, self synchronizing serial bit stream, suitable for transmission on the coaxial cable. The inputs to the Manchester encoder are a 10 MHz clock and the output of the TX shifter. The Manchester encoder is controlled by the transmit state machine, and collision jam.

3.4.3.2 Transmit Enable Sync -- The TX enable sync logic enables the transmission of data when either the transmission slottime counter has expired and at the end of an interpacket delay.

TX enable sync is controlled by the transmit state machine and collision jam.

3.5 TRANSMIT SECTION

The transmit section of the link module prepares data for transmission onto the ETHERNET. After transmission, this logic will report status on the data transmitted. In order to accomplish this, the transmit section performs the following functions.

- Buffering of transmit data and status information between the host processor and the physical channel
- Parallel-to-serial data conversion
- Preamble generation
- CRC generation

The following paragraphs explain the functional sections of the transmit logic.

3.5.1 Data Section (Link Memory Buffers)

The link memory transmit buffer is made up of two sections, the data section and the status section.

There are two ways that the link module interacts with a link memory transmit buffer.

- Data Section -- This is the data being transmitted. It is written by the port and read by the link.
- Transmit Status Information Section -- Upon successful completion of transmission of a frame or after 16 unsuccessful attempts to transmit a frame, the link will write status information to the link memory transmit buffer.

Figure 3-5 shows the format of the transmit buffer before transmission and Table 3-8 gives a description of the buffer bits.

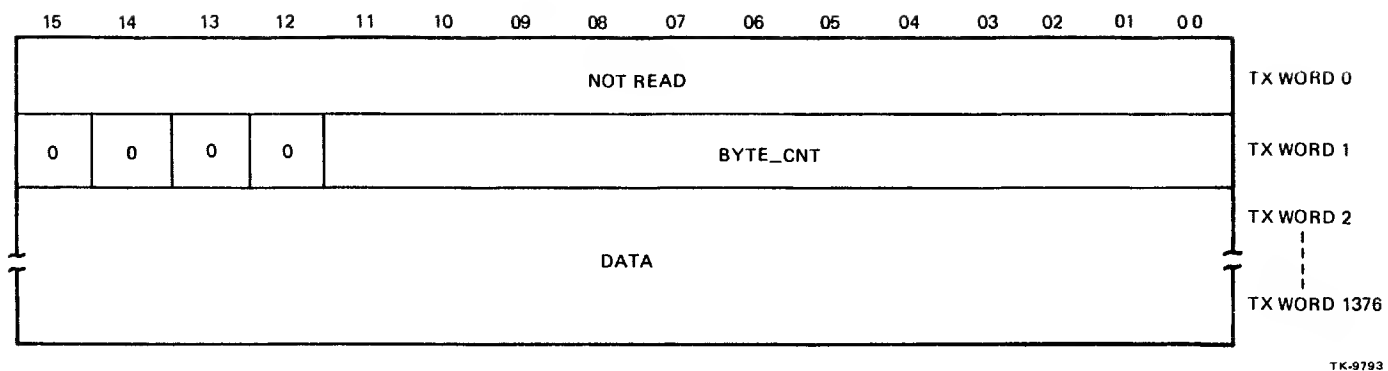


Figure 3-5 Transmit Buffer Format Before Transmission

Table 3-8 Transmit Buffer Bit Descriptions

Word	Bits	Field	Description
TX Word 0	<15:00>	Not Read	
RX Word 1	<11:00>	BYTE CNT	Transmit Byte Count register. Written and cleared by the port.
TX Word 2- TX Word 1376	<15:00>	Data	Written by the port. When transmitting an odd number of bytes, data found in bits <07:00> in the last entry location of the buffer is sent last.

3.5.2 TX Data Latch

The TX data latch is used to transfer transmit data from the link memory data bus to the TX shift multiplexer. The TX data latch is controlled by the transmit state machine and link memory bus controller.

3.5.3 TX Message Byte Counter

The TX byte counter is implemented as a 12-bit counter that is loaded by the transmit state machine from information contained in the link memory buffer. The TX byte counter contains the number of data bytes to be transmitted over the physical channel and is decremented to zero by 10 MHz clock. The count output of the TX byte count register is an input to the transmit state machine.

3.5.4 TX Frame and Byte Sync

The TX frame and byte sync signals provide a 100 ns pulse signal every 16- and 8-clock periods respectively. The TX frame and byte sync signals are implemented as an UP counter and a terminal count detect circuit. These signals are initialized by the transmit state machine. During the odd byte case TX frame is advanced eight bits just before sending the four byte CRC. TX frame and byte sync are controlled by the 10 MHz clock and the TX byte count register.

3.5.5 TX Shift MUX

The TX shift multiplexer is used to selectively transfer a 16-bit word of either preamble or transmit data to the TX shifter. The TX shift multiplexer is controlled by the transmit state machine.

Sel 1	Output
Sel TX Data L=0	Transmit Data
Sel TX Data L=1	Preamble Data

TX Multiplexer Selection Chart

3.5.6 TX Shifter

The TX shifter converts parallel data from the TX shift multiplexer into a serial output data stream that goes to the CRC generator and the Manchester encoder. The TX shifter is parallel loaded and is controlled by the TX clock and the transmit state machine.

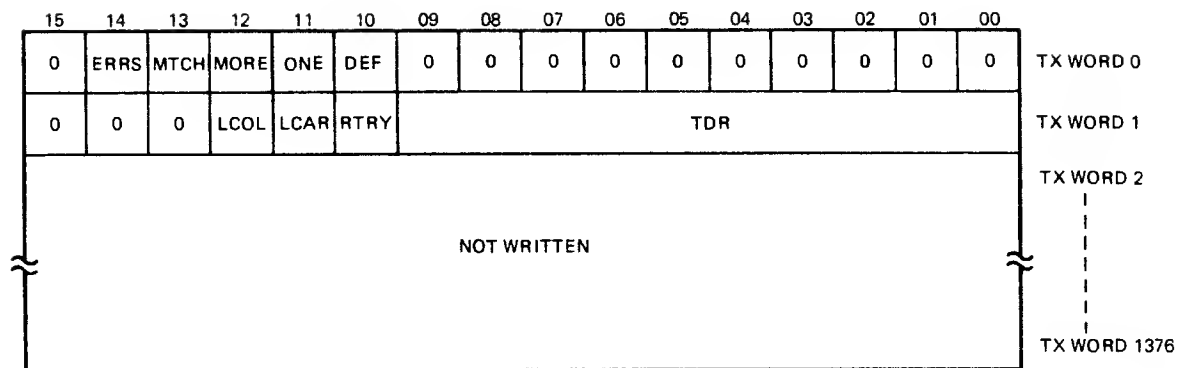
3.5.7 TX Output MUX

The TX output MUX is used to select the output of the TX shifter or the CRC generator for output to the transmitter.

3.5.8 TX Status Information

The transmit status information is written into the link memory transmit buffer by the link either after a successful attempt to transmit a frame or after 16 attempts to transmit a frame have failed. The first two words of the transmit buffer are used to store this information. Figure 3-6 shows the format of these words in relation to the rest of the buffer. Table 3-9 describes each of the status bits.

For information about the transmit data buffer before transmission, refer to paragraph 3.5.1 of this chapter.



TK-9796

Figure 3-6 Transmit Buffer Format

Table 3-9 Transmit Status Bit Descriptions

Word	Bits	Field	Description
TX Word 0	<14>	ERRS	Error Summary - The logical OR of LCOL, LCAR, or RTRY was set. Written and cleared by the link.
TX Word 0	<13>	MTCH	Station Match - Set by the link when the destination address of the message matches one of the addresses of the UNA.
TX Word 0	<12>	MORE	Multiple Retries Needed - Set when more than one and less than 16 retries were needed to transmit a frame. Written and cleared by the link.
TX Word 0	<11>	ONE	One Collision - Set when exactly one retry was needed to transmit a frame. Written and cleared by the link.
TX Word 0	<10>	DEF	Deferred - Set when the transmitter experienced no collisions but had to defer while trying to transmit a frame. Written and cleared by the link.
TX Word 1	<12>	LCOL	Late Collision - A collision has occurred after the slot time of the channel has elapsed. Written and cleared by the link.
TX Word 1	<11>	LCAR	Loss of Carrier - Carrier was either not present on the channel during transmission or transceiver power was not present. Written and cleared by the link.
TX Word 1	<10>	RTRY	Retry - Transmitter has failed in 16 attempts to transmit the frame due to collisions on the medium. Written and cleared by the link.
TX Word 1	<9:0>	TDR	Time Domain Reflectometry Value - Valid only when RTRY or LCAR is set. Written and cleared by the link. All ones indicates an overflow condition.

Table 3-9 Transmit Status Bit Descriptions (Cont)

Word	Bits	Field	Description
TX Word 2-	<15:00>	NOT WRITTEN	
TX Word 1376			

3.5.9 Transmit State Machine

The transmit state machine controls are:

- The link data path during transmission, and
- The access of the buffers in the link memory.

The transmit state machine is implemented in PALs and consists of the following states:

1. Transmit Enable State - Entered by the transmit state machine when the port asserts TATT. Exited after carrier has gone away and the interpacket gap timer has elapsed.
2. Preamble/Start Bit - Entered after the transmit enable state. The preamble consists of 64 bits of alternating ones and zeros ending in a double one. The preamble is loaded into the TX shifter as four 16-bit words to be shifted serially out onto the wire.

If the transmitter is enabled and there are no collisions on the wire, the transmit state machine will increment the TX pointer and then load the transmit byte count during the loading of the first word of preamble.

3. Data State - Entered after the fourth word of preamble is loaded into the TX shifter. During this state, data is transferred from the link memory data bus to the TX shifter to be serially shifted onto the wire. This state remains active until the TX byte count register has expired or a collision occurs.
4. CRC State - Entered after the data state if the DTCR bit is not set and exited after 32 bits of CRC are transmitted or a collision occurs.
5. Write Status - Entered after the CRC state. During this state the transmit state machine writes the transmit status into the link memory buffer residing on the port. If there are no collisions and no collision errors, then the transmit state machine resets the TX pointer, write status Word 0, and write status Word 1.

6. Retry - Entered if there is a collision on the wire. During this state the transmit state machine continues transmitting, a process known as jamming, for 32-bit times. At the end of enforcing the jam, the transmit state machine delays for attempting to retransmit again. This delay is based upon some multiple number of slot times. This state is further described in the RETRY section.
7. Done.

3.6 RETRY LOGIC

The retry logic controls the scheduling of the retransmission of packets when a collision has occurred. This logic uses the binary exponential backoff algorithm. Basically the algorithm waits a generally increasing random number of slot times before retransmission. The random number must be between 0 and 2^K , where K is the $\min(n, 10)$ for the nth transmission.

3.6.1 Collision Jam

Collision jam keeps the transmitter on for 32-bit times after a collision is detected and the preamble has finished transmitting.

Collision jam is asserted by the leading edge of collision detect and is used as an input to the retry slot time counter, the carrier multiplexer, and the TX enable sync.

3.6.2 Slot Time Counter

The slot time counter is a 51.2 microseconds modulus counter. The slot time counter begins its count upon recognition that the retry state machine is in the backoff state. The output of the slot time counter is used as an input to the retry interval counter.

3.6.3 10 MHz Oscillator

The 10 MHz oscillator is implemented as an RC voltage controlled oscillator. The oscillator provides the clock for the random number generator.

An RC oscillator is used so that the probability of the retry logic of other nodes on the ETHERNET becoming synchronized is decreased.

3.6.4 Random Number Generator

The random number generator is implemented as a 10-bit binary counter that continuously counts from power-up and is never reset. The 10 outputs of the random number generator are the inputs to the random interval mask/latch.

3.6.5 Random Interval Mask/Latch

The random interval mask is combinational logic which masks out bits in the random number according to the number of retries needed to successfully transmit a packet. The mask ensures that the random number is between 0 and 2^K , where K is the $\min(n, 10)$ for the nth transmission. Inputs to the random interval mask are

the 10 output lines from the random number generator, and the retry counter. The output from the mask is latched into the random interval latch.

3.6.6 Interval Counter

The retry interval counter is a binary counter that counts the number of slot times that have elapsed. Counting ceases when the number of slot time intervals is equal to the number of random slot times provided by the random interval mask/latch.

3.6.7 Retry Counter

The retry counter counts the number of retransmissions that have occurred. The retry counter is incremented by the interval counter and reset by the transmit state machine. The outputs of the retry counter are the inputs to the transmit status register, and the retry interval mask.

3.6.8 Retry State Machine

The retry state machine, not shown on the block diagram, is used to control the retry process during a collision. The retry state machine is implemented in a PAL and consists of the following states.

1. Jam State - This state is entered if a collision is encountered during transmission of data on the wire. During this state, the transmit section remains transmitting for 32-bit times if the collision occurred during the data state. If, however, the collision occurred during the preamble state, the transmitter will continue transmitting the preamble and then jam for 32 bits. During the jam state, the CRC is disabled.
2. Backoff State - This state is entered after the jam state.

At the end of enforcing jam, the transmitter delays before attempting to retransmit again. This delay is an integral multiple of slot times. The number of slot times to delay before the n th retransmission attempt is chosen as a uniformly distributed random integer r in the range of $0 \leq r \leq 2^k$ where $k = \min(N, 10)$. If all 16 attempts to transmit fail, the event is reported back as a RTRY error.

3. Force Collision - This is a maintenance self-test function and is valid if the loop and COLL bits in the mode register are both set and the function is reset by clearing these bits.

Force test allows the microprogrammer to single step through the collision retry algorithm one attempt at a time by simulating a collision on the wire without being physically linked to it. Each attempt to transmit forces a collision internally to the link module. The transmit

state machine then goes through the collision jam and retry states. The retry counter is then incremented and the transmit state machine then writes the appropriate status information to link memory.

3.6.9 Time Domain Reflectometry

The TDR counter is ten bits wide modulus counter. It is cleared by the transmit state machine and counts upon the recognition of carrier during transmission. Counting ceases either due to a collision, loss of carrier, or if it has reached its modulus. The value of the TDR is written into memory by the microprocessor. TDR is used to determine the location of suspected cable faults.

3.7 RECEIVE SECTION

The receive logic on the link module is used to:

- Convert serial data to parallel data
- Count the number of bytes received
- Write the received data into the link memory buffers
- Write status information and message length into the receive buffers

3.7.1 Data Section (Link Memory Buffers)

The link memory receive buffer is located in link memory and is written only by the link. It contains the data and status information provided by the physical channel and the receiver state machine. Figure 3-7 shows the format of the link memory receive buffer. Table 3-10 describes the status and data bits of the buffer.

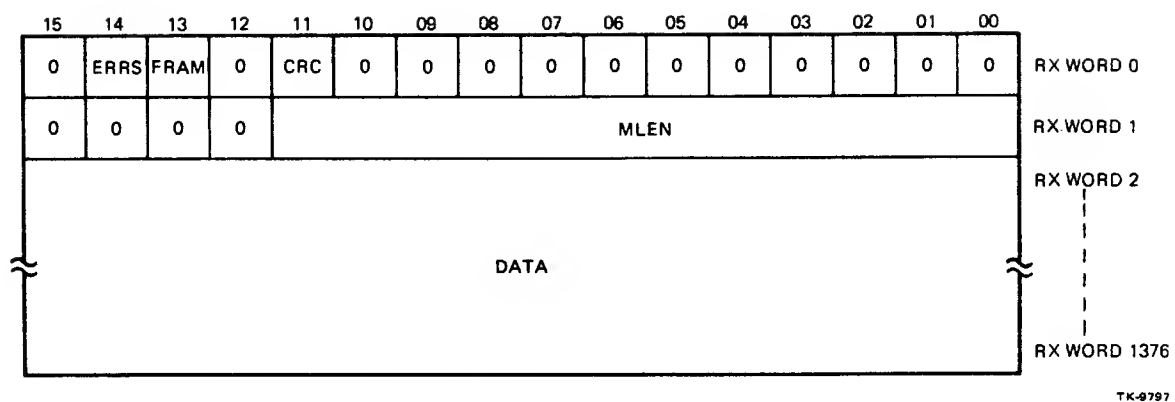


Figure 3-7 Receive Buffer Format

Table 3-10 Receive Buffer Status and Data Bit Description

Word	Bits	Field	Description
RX Word 0	<14>	ERRS	Error Summary - The logical OR of FRAM, CRC. Written and cleared by the link.
RX Word 0	<13>	FRAM	Frame Error - Indicates that the incoming frame contained a non integer multiple of 8 bits and the CRC value at the last 8-bit boundary was in error. Written and cleared by the link.
RX Word 0	<11>	CRC	Cyclical Redundancy Check - Frame check error, data is not valid. Written and cleared by the link.
RX Word 1	<11:0>	MLEN	Receiver Byte Count Register - Written by the link. This register latches at all ones indicating a babbling node on the network or broken byte count detect logic.
RX Word 2- RX Word 1376	<15:00>	DATA	Written by the link. During the odd byte case, data would be found in bits <7:00> of the last word written.

3.7.2 Receive MUX

The RX multiplexer is used to select data from the ETHERNET or the output of the TX output MUX. The RX multiplexer is controlled by a loopback signal.

Sel 1	Output
Loopback=0	Receive Data
Loopback=1	Transmit Data

Multiplexer Output Selection Chart

3.7.3 Receive Shifter

The RX shifter is a 16-bit wide device that frames the incoming serial bit stream into a word stream. A normal reception sequence consists of the continuous shifting of the alternating "ones" and "zeros" that comprise the preamble through the shifter. Upon the recognition of the double "one" pattern that indicates the start of data, the data is then framed into the RX shifter.

The RX shifter is controlled by the receive state machine and RX clock. The output of the RX shifter is transferred to the RX data latch.

3.7.4 RX Data Latch

The RX data latch is used to transfer receive data from the RX shifter to the link memory data bus. The RX latch is implemented as three 8-bit counters whose inputs are the outputs of the RX shifter. Two of these latches work together to transfer data found on 16-bit boundaries to the link memory data bus. The third latch, strobed every 8 bits, is used during the odd byte case to transfer the last byte of data to the link memory data bus. Data strobed into the RX latch is transferred onto the link memory data bus using the handshake provided by the link memory bus controller. The RX latch is controlled by the receive state machine, carrier, bit 0 of the RX byte count register (to detect odd bytes), and the link memory data bus control logic.

3.7.5 RX Frame and Byte Sync

The RX frame and byte sync signals provide a 100 ns pulse signal every 16- and 8-clock period respectively. The RX frame and byte sync signals are implemented as an up counter and a terminal count detect circuit. These signals are initialized by the recognition of start bit. The frame and byte sync are further gated with RCLK L to minimize skew.

3.7.6 RX Byte Counter

The RX byte count is implemented as a 12-bit counter that may be accessed over the link memory data bus. The RX byte counter contains the number of data bytes that are received from the physical channel.

The RX byte counter is incremented by RX clock and is controlled by the receive state machine. The counter will latch up to all ones for an overflow condition. The output of the RX byte counter is passed to the link memory data bus by the TX/RX status multiplexer.

3.7.7 Receive State Machine

Control of the link data path during reception is provided by the receive state machine. The receiver state machine is implemented in PALs and consists of the following states.

1. Receiver Enabled - Entered by the receiver state machine upon setting the on bit in the command register, or upon completing the transfer of an incoming frame, or after the bad packet state, or after the miss state. The receiver state machine stays in this state until carrier is no longer present on the wire.
2. No Carrier - Entered after receive enabled state when carrier is no longer present on the wire and exited when the carrier signal comes up.

3. Carrier - Entered after the no carrier state upon presence of carrier on the wire. During this state the receive state machine looks for a valid preamble, a free receiver buffer, and checks for a runt, no address match, or a start bit.
4. Pointer Reset - Entered after the no carrier state upon presence of carrier on the wire. This state resets the receiver address pointer on the link memory. Exited after one clock period.
5. Pointer Increment - Entered after the pointer reset state. This state increments the receiver address pointer on the link memory to point to the data section of the buffer.
6. Data Request - Entered during the carrier state after recognition of a valid start bit. Exited upon loss of carrier and a bad packet, or status write and a valid packet. During this state, the receive state machine transfers data from the wire to link memory and increments the RX pointers.
7. Bad Packet - Entered after the carrier state if either the packet was less than 64 bytes (runt packet) or the packet did not pass address recognition.
8. Valid Packet - Entered after the carrier state if the packet passed address detection was not a runt packet and there was a free receiver buffer to put the packet in.
9. Miss - Entered after the carrier state if the packet passed address detection, was not a runt packet, and there was no free receiver buffer available.
10. Write Status - Entered after the valid packet state. During this state, status information is written to the link memory buffer.
11. End of Reception - Entered after the write status state. Exited after one clock period.

3.7.8 Interpacket Delay

The interpacket delay prevents the transmission of data for at least 9.6 microseconds after the last carrier detect.

The interpacket delay is asserted by the trailing edge of carrier and is used as an input to the transmit state machine.

3.8 STATION ADDRESS DECODE

The station address detect logic checks the destination address of the incoming packet to determine if the packet is addressed to this node. A packet passes address detection if at least one of the following is true:

1. Logical address match: the destination address of the packet exactly matches one of the 11 possible logical addresses of the node.
2. Physical address match: the destination address of the packet exactly matches the physical address of the node.
3. Promiscuous mode: this mode accepts all packets regardless of the destination address.
4. Enable all multicast: this mode accepts all packets with multicast address regardless of the destination address.

The station address match is then used by the receive state machine. This signal is synchronized to the 10 MHz system clock by a dual rank synchronizer before entering any logic operating off the system clock.

3.8.1 Physical/Logical Address Detection

Physical/logical address detection is done by serial comparing each bit of the destination address on the wire against the contents of the 48*12 station address RAM. The serial compares of the physical and logical addresses are all done in parallel and are enabled by the receiver state machine.

The physical/logical address is written into the station address RAM by 48 sequential memory writes over the link memory data bus.

3.8.2 Promiscuous Mode

In this mode the receiver logic will accept all packets that are sent, regardless of the destination field of the packet.

3.8.3 Enable All Multicast

This mode accepts all packets with multicast addresses regardless of the destination address.

3.9 CRC LOGIC

The CRC logic implements the 32-bit CRC using the AUTODIN-II polynomial as the generating polynomial. The generation and checking of the CRC is done using a 32-bit register implemented in PALs which acts as a shift register, XOR gates, and combinational logic for control.

CRC logic is half-duplex during transmission, reception, and loopback. During loopback the CRC logic is dedicated to the transmit section of the link unless DTCR is set in the link mode register. (If DTCR is set, the CRC logic is dedicated to the receiver.)

For checking the CRC at the end of a packet, a residue detector is used to monitor the data as it shifts through the CRC generator. The residue detector is strobed on 8-bit boundaries. If there are no CRC errors, the output of the CRC to the residue detector is the value of:

11000111 00000100 11011101 01111011

(Where the leftmost bit corresponds to the X^{*31} term of the polynomial and the rightmost to the X^{*0} term.) Any other value indicates an error.

The input to the CRC generator is either the transmit data stream or the receive data stream. The CRC generator/checker is controlled by the transmit state machine, receive state machine, RX clock, TX clock, and loopback.

NOTE

Output of the CRC PALs are asserted "low".

3.10 TX/RX STATUS

The TX/RX status multiplexer is used to transfer status information from TX Word 0, TX Word 1, RX Word 0, or RX Word 1 to the link memory data bus for writing into the appropriate link memory buffer that resides on the port. The TX/RX status multiplexer is enabled and controlled by the link memory data bus control logic and the receive and transmit state machines.

Sel 1	Sel 0	Output
0	0	TX Status Word 0
0	1	TDR, TX Status Word 1
1	0	RX Status Word 0
1	1	RX byte count

TX/RX Status Multiplexer Selection Chart

3.11 LINK MEMORY

The link memory section is the part of the link module that communicates with the port module. This section contains 16K words of RAM which is used by the link module to buffer packets that are to be received or transmitted on the ETHERNET. This 16K of memory is broken down into sixteen 1536 byte buffers. The first four bytes of each buffer are used to convey status information about the packet.

Addressing of link memory is provided by the port module over one of two over-the-top cables connecting the port to the link.

This memory is arbitrated for and accessed by four different processes:

1. Link transmit state machine.
2. Link receive state machine.
3. DMA engine (described in the UNA Port Module Functional Description).

4. T11 (described in the UNA Port Module Function Description).

The link memory arbitrator resides on the port module.

3.12 LINK MEMORY BUS CONTROLLER

The link memory bus controller is a simple state machine that provides the necessary handshake involved in transferring data between link memory and the transmitter or receiver.

4.1 OVERVIEW

The microcode provides the microcode instructions necessary to control the T11 microprocessor contained on the port module. This code in conjunction with the T11 is responsible for data encapsulation and decapsulation, data link management, and all channel access functions. This allows for maximum data throughput with a minimum of intervention by the host processor.

In order to understand how the microcode of the DEUNA functions, it is necessary to understand how the DEUNA is programmed. Information on how the DEUNA is programmed can be found in Chapter 3 of the DEUNA User's Guide (EK-DEUNA-UG).

4.2 STRUCTURE

The microcode of the DEUNA is structured as a series of concurrent, cooperating processes that are executed under the control of a supervisor program. These processes are created at the time the DEUNA is powered up and are entirely self-contained. Each process is capable of performing its specific function without assistance from any other process.

4.3 SUPERVISOR

The supervisor is made up of the routines that are needed to:

- Control the scheduling of the different processes used in the DEUNA, and
- Maintain the status and data needed for the operation of the DEUNA.

There are two different types of routines executed by the supervisor, interrupt routines and subroutines.

1. Interrupt Routines -- These routines are executed as a result of a specific interrupt generated by the hardware of the DEUNA. These routines will normally run to completion at the level of the interrupt.
2. Subroutines -- These routines are called by a specific process while that process is running. These routines are accessed by way of a dispatch table contained in ROM. This table is written into the WCS of the DEUNA during initialization.

4.3.1 Initialization

The initialize routine is the first supervisor routine to be executed after the completion of self test.

The function of the initialize routine is to:

1. Reset the hardware of the DEUNA to a known state;
2. Build the supervisor dispatch tables in Writeable Control Store (WCS);
3. Create the data structures in WCS required by the micro-code;
4. Clear all the internal counters, the multicast list, mode register, and descriptor ring lengths;
5. Load the physical and broadcast address into the station address RAM on the link module;
6. Enable all hardware interrupts;
7. Load the address of the receive buffers and allocate the transmit buffers; and
8. Start the null process (this executes at priority zero).

4.3.2 Scheduling

The supervisor performs the scheduling of processes through the use of a request mask. When the T-11 receives an interrupt requesting a particular process to be run, the interrupt service routine sets a bit in the request mask. The next time the null process runs it will scan the request mask to see if any low priority processes are scheduled to be run.

All the processes will execute at the CPU priority of zero with the exception of the datagram receive process. As a result there is no context switching between low priority processes. This means that each process, with the exception of the datagram receive process, will run to completion before the request mask is scanned again. The receive process runs at the priority of the hardware interrupt.

When a process has completed it will return to the supervisor by executing an RTI instruction or calling the supervisor command complete routine.

Table 4-1 gives a list of the processes and the order of execution (priority).

Table 4-1 Priority of Processes

Process	Priority
Datagram Receive	1
Port Command	2
Timer	3
Loop and Maintenance	4
Datagram Transmit	5
Null	6

4.3.3 Datagram Receive Process

The datagram receive process is used to transfer receive datagrams from the receive buffers on the link to host memory. This process is the highest priority process because it has the greatest impact on the throughput of the Ethernet and the DEUNA.

The receive process is started by the buffer filled interrupt or by the START port command. The process is ended when:

1. The datagram was written into host memory
2. Status information was written into the descriptor
3. A new buffer descriptor was read from the ring entry

The receive process executes at a hardware priority level of five and can only be interrupted by DMA done, power failure, or errors. Because the amount of processing performed is short (get buffer, start DMA machine), it is possible for other processes to run between the time the DMA machine is started and the DMA done interrupt is generated.

The receive process is initiated in two ways:

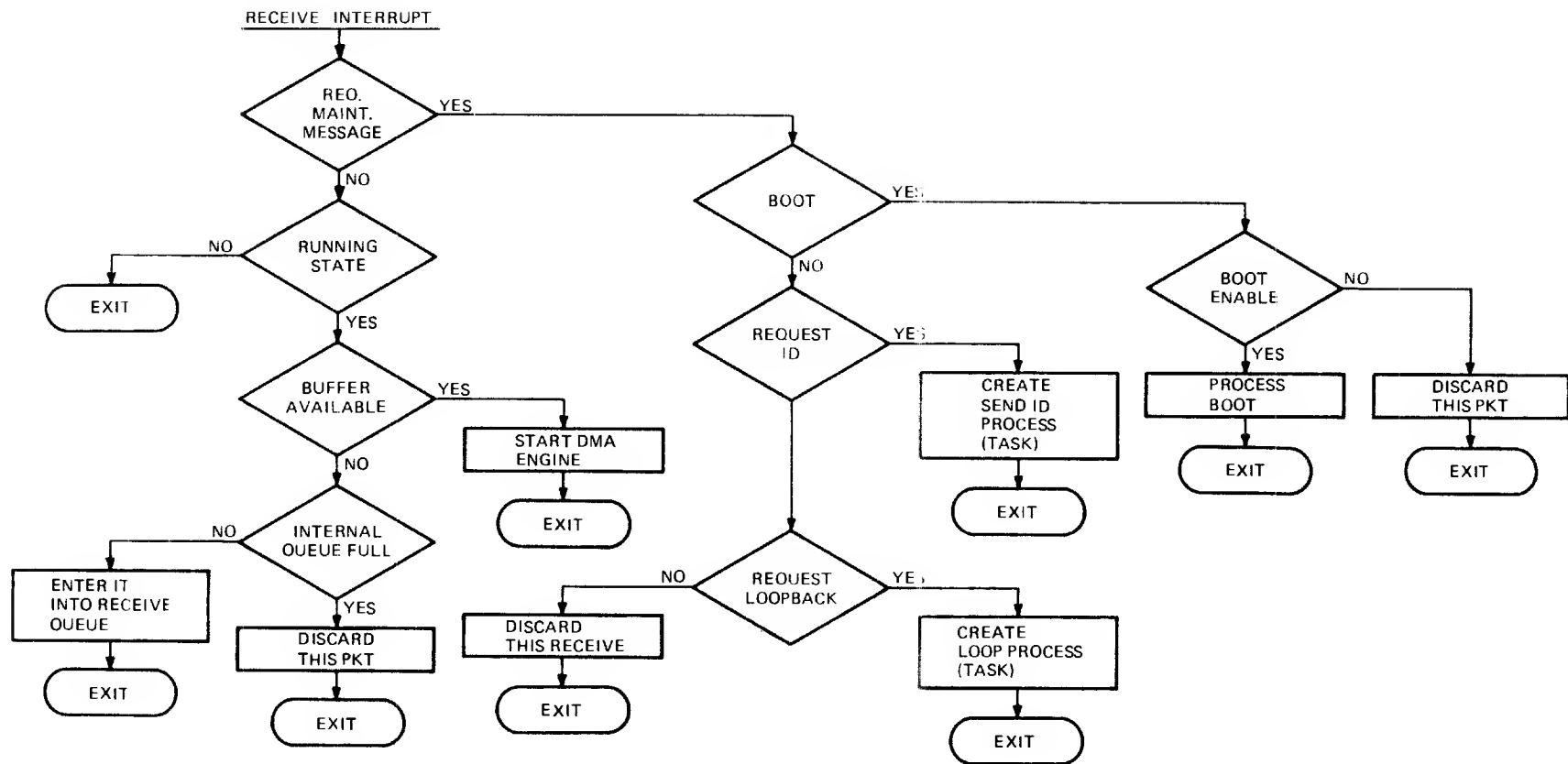
- A datagram was received and an interrupt was sent to the T-11.
- A poll demand or start command was received from the host. Either causes an interrupt to generate, and the receive process to start.

The receive process performs the following:

1. Poll receive ring to get a buffer in host memory.
2. Load and start DMA machine.

3. When DMA is done, execute an RTI instruction or run the null process.

Figure 4-1 and Figure 4-2 show the function of the microcode for the receive process.



TK-9785

Figure 4-1 Receive Flow Diagram

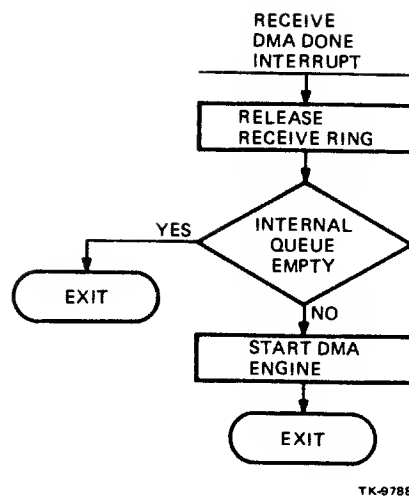


Figure 4-2 Receive DMA Done Flow Diagram

4.3.4 Command Execution Process

The command execution process is used to receive commands from the host processor. The host sends commands to the DEUNA via a structure in host memory called the port control block (PCB). The host tells the DEUNA that it has placed a command in the PCB by writing to PCSR0. This causes an interrupt to be generated. When the interrupt is received by the T-11, the supervisor will read the command from the PCB and schedule the requested process for execution.

The command process reads the low byte of PCSR0 and uses the code in bits <03:00> to select one of the port command routines.

Figure 4-3 shows the different command processes.

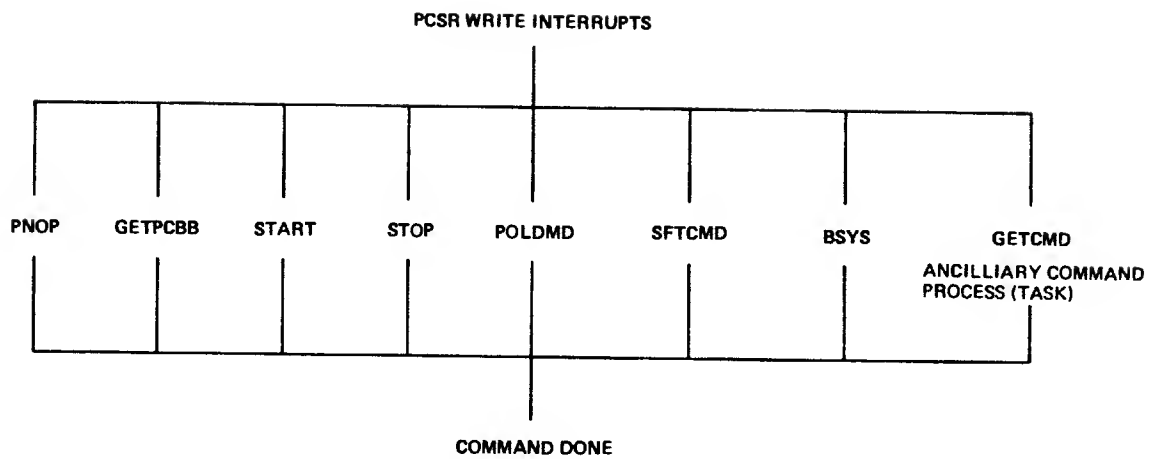
4.3.4.1 Port Commands -- The following port commands are used by the DEUNA.

1. Get PCBB -- The DEUNA reads the address of the PCBB from PCSR2 and PCSR3 and stores it in the WCS.
2. Get CMD -- Requests execution of the ancilliary command process.
3. Self-Test -- Invokes the internal ROM based diagnostic feature of the DEUNA. All datagram activity are aborted and the DEUNA returns to the ready state.
4. START -- The transmit and receive processes are activated and the ring pointers are reset to the base of the rings.
5. BOOT -- The UNA enters the primary load state and requests a program from the load server address.
6. POLL Demand -- The transmit and receive processes are activated if not already active. The transmit and receive rings in host memory are polled.
7. STOP -- The DEUNA completes the current transmit and receive operations and does not fetch any more ring entries until a START command is received.

This command is implemented by:

- a. Clearing the status flag that indicates the DEUNA is in the running mode, and
- b. Setting the state of both rings to inactive.

This also causes any datagrams in the link memory buffers to be lost.



TK-9789

Figure 4-3 Port Command Processes

4.3.4.2 Ancilliary Commands -- The ancilliary commands are sub-routines that are called by the get CMD port command. Each of the subroutines executes its specific task and then exits to:

- The command done supervisor routine, or
- Set an error flag and call the appropriate function error routine.

The ancilliary commands executed by the DEUNA are as follows.

1. No Operation (NOP) -- Calls the command done routine.
2. Load and Start at Address (LDSTA) -- Executes a JSR PC instruction directly to the address specified by the PCB.
3. Read Default Physical Address (RDEFPA) -- The physical address contained in ROM on the port module is written to the PCB.
4. Write Physical Address (WRTPA) -- The physical address specified by the PCB is placed in the location reserved for the current physical address (PHYADR). The formatting routine is called to build the data format needed by the address filter of the link module and the data is loaded into the link.

The link must be halted to execute this command.

5. Read Physical Address (READPA) -- The current physical address is written to the PCB.
6. Write Multicast List (WRTMLT) -- The multicast list is read and stored in a table in WCS. This list along with the broadcast and physical addresses is formatted and written into the address filter in the link.

The link must be halted to execute this command.

7. Read Multicast List (RDMLT) -- The multicast list is written to the UNIBUS data block specified by the PCB.
8. Read Ring Format (RDRFMT) -- The ring format block of the DEUNA is written to the UNIBUS data block specified by the PCB.
9. Write Ring Format (WTRFMT) -- The ring format is read from the UNIBUS data block and written into WCS of the DEUNA. To maximize performance, the address of the last entry in each ring is calculated. These addresses along with the length of the rings in bytes are saved. The address of both of the rings is written into the ring descriptor for the next entry to be fetched from each of the rings (receive and transmit).

The DEUNA can not be in the running state when this command is executed.

10. Read Counters (RDCNTR) and Read and Clear Counters (RCLCNT) -- The counters that are maintained in WCS are written to host memory. If the command is a read and clear command, the counters are read and then cleared.
11. Dump Internal Memory (DMPMEM) -- A block of data contained in the memory of the DEUNA is specified by the command and transferred to a data buffer in host memory.
12. Load Internal Memory (LDMEM) -- A specified block of data in host memory is copied into the memory on the DEUNA.
13. Read/Write System ID Parameters (RDPARM), (WTPARM) -- The system parameters list is copied from either:
 - A data buffer in host memory to the DEUNA, or
 - The DEUNA to a data buffer in host memory.
14. Read Load Server Address (RDSERV) -- The load server address currently in use by the DEUNA is written into the PCB.
15. Write Load Server Address (WTSERV) -- The load server address in the PCB is written to the DEUNA.

4.3.5 Timer Process

The timer process is executed every second in response to an interrupt generated by the timer on the port module of the DEUNA. The timer is used to:

1. Send an ID message to the ETHERNET every 10 minutes,
2. Keep track of seconds since the counters maintained by the DEUNA were last zeroed. This keeps track of activity in the DEUNA, and
3. Provide timing for various boot operations.

4.3.6 Loop and Maintenance Process

The loop and maintenance process is used to loop data back onto the network, send system ID messages, and perform system boots. The processes are handled as follows:

1. Loop Messages -- Loop service is provided by the microcode to verify that the DEUNA is properly connected to the network and is able to receive and transmit messages.

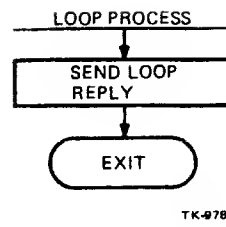


Figure 4-4 Loop Process Flow Diagram

The microcode screens the received messages, in internal memory, to see if there are any loop type messages. (All messages that are not loop type messages are handled as normal datagrams.) If a loop type message is found and is error free it is handled as follows:

- a. The microcode modifies some of the address fields.
- b. Places the receive buffer into a transmit buffer.
- c. Transmits the message.
- d. The receive buffer is returned to the receive free buffer queue.

These type of messages are not passed to the host for processing by higher level software.

Figure 4-1 and Figure 4-4 show the function of the microcode for the loop process.

2. System Identification Messages -- When enabled, the microcode will build and transmit a system identification message. This message is transmitted to the network every 8 to 10 minutes to identify the node to the network. This address is also sent if a request station ID message is received. A request station ID message is not processed as a datagram.

Figure 4-1 and 4-5 shows the function of microcode for the system ID process.

3. Boot Messages -- When enabled, the microcode monitors the incoming receive messages for a boot message. If a boot message is received, the following action takes place:
 - a. Datagram service is turned off,
 - b. A request program load message is sent to the requesting station, and
 - c. The WCS is down-line loaded and program execution is started out of the WCS.

This procedure may be used to load remote console code or to load the system secondary loader. If the system is to be booted, as determined by the boot message, the microcode will halt the system by asserting ACLO and starting the power fail sequence before it transmits the program request message.

Figure 4-1 shows the function of the microcode for the boot functions.

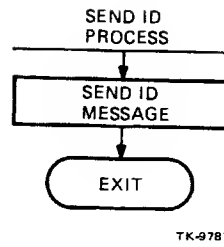


Figure 4-5 Station ID Flow Diagram

4. Power-up Boot -- If the microcode is enabled to do a power-up boot, the microcode will:
 - a. Halt the system,
 - b. Start power fail sequence,
 - c. Transmit program request message, and
 - d. Wait for secondary loader.

If the system boot port command is received, the microcode will handle the request the same way except it does not halt the system.

5. Remote Boot -- For a remote boot from the system ROM (not located on DEUNA), the microcode asserts ACLO.

4.3.7 Transmit Datagram Process

The function of the transmit process is to read a datagram located in host memory and load it into a buffer in link memory for transmission onto the ETHERNET.

The transmit process is activated when the DEUNA receives a poll demand and will be deactivated when the DEUNA comes to a ring entry that is not owned by it.

The transmit process functions as follows.

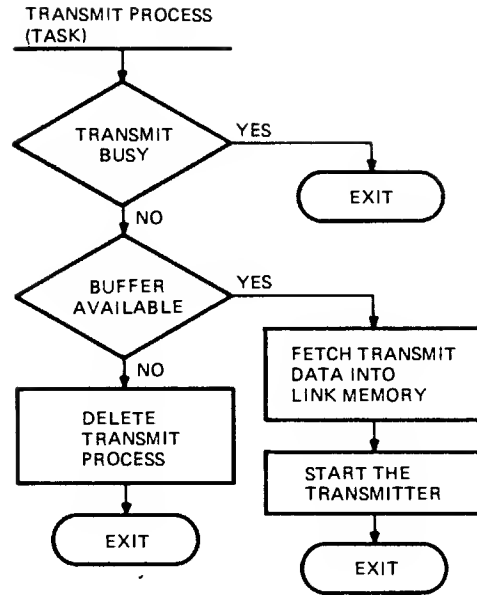
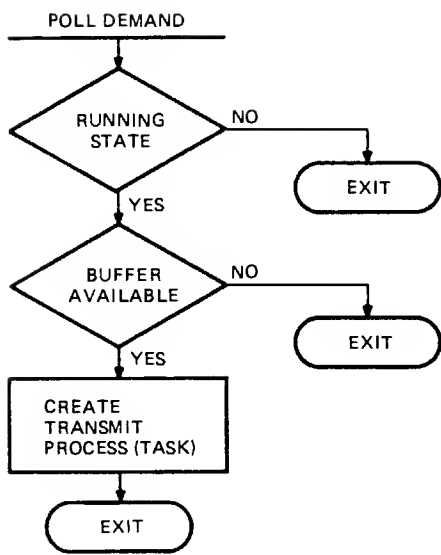
1. A poll demand or start command generate an interrupt which starts the transmit process.
2. A transmit buffer in link memory is allocated.
3. A ring entry is fetched from the host and is stored in the transmit descriptor of the DEUNA called NEXT.
4. The data described by the ring entry is loaded into the transmit buffer in link memory.
5. The link is given the address of the buffer to be transmitted on the ETHERNET.
6. The link transmit function is started.
7. The ring descriptor in the DEUNA is renamed CURRENT.
8. When the link has finished transmitting the buffer, a transmit done interrupt is generated.
9. The transmit status from the link is stored in the ring entry addressed by the CURRENT ring descriptor in the DEUNA.

10. The ring entry is released and the CURRENT descriptor is marked empty.
11. A return is executed and the NULL process will run. If the transmit process is still the highest process in the request mask the transmit ring will be polled and the process repeated.

Figure 4-6 and Figure 4-7 shows the function of the microcode for the transmit process.

4.3.8 Null Process

The null process scans the request mask to see if any low priority process is scheduled to run. All the low priority processes run sequentially. Each process runs to completion before the request mask is scanned again.



TK-9790

Figure 4-6 Transmit Flow Diagram

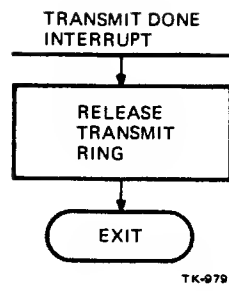


Figure 4-7 Transmit Done Flow Diagram

